Introduction

Wishlist is a smart shopping app that helps you make informed purchasing decisions by comparing prices from different online retailers. With its user-friendly interface you can easily find the best deal on whatever you're looking for. Whether you're shopping for clothes, electronics, or home goods, *Wishlist* gives you all the information you need to make a smart purchase.

It not only provides you with the ability to compare prices from multiple online retailers, but it also keeps a database of all the products you might buy in the future and helps you maintain a '*Wishlist*'. This database can be constantly updated by adding new products or deleting old ones, making it easy for you to stay organised.

With another feature 'Discover Latest in TECH', you can easily discover new products, see what's popular, and stay on top of the latest trends in the world of Tech and online shopping.

Wishlist is the perfect tool for those who spend hours scrolling through shopping websites, searching for the best deals and the right products. It is a one stop go to take care of all your shopping needs.

Using this app makes your shopping experience smooth and effortless and saves you a lot of time and effort and above all, money!

Related Studies

We studied many python modules, apis, libraries and finally settled for tkinter to use as a building tool for front end with python as back-end.

Similar applications can be built by using flask or Django on the back end and HTML, CSS and JavaScript on the front-end. Flask and Django can take values you type in the input field and run your corresponding python function to show you search results.

Like tkinter, it can also be used to connect to a database and store and retrieve values from there.

For the graphical user interface, Tkinter was used. It is an easy-to-use python library that can help us make small applications faster and hassle-free.

If the *Wishlist* were made with the use of flask or Django, it could be converted into a portable executable package using electron.

For web-scraping, selenium, scrapy and autocraper can be also used. Among these, autoscraper is the easiest one to use and has the shortest code as in it you can build a model of the search results you want for one entity, and it smartly detects a pattern and returns you all similar such cases on the webpage.

For example, you make a model for scraping the price, description and ratings of one product and it returns the result for all products on the page.

SQLite is another substitute for maintaining databases which comes in-built with python; however, it cannot be used to maintain large databases like we can with MySQL.

Figma can be used to design beautiful UI/UX for our application using its simple interface by dragging and dropping elements/widgets as you please and adjusting the colours, font, etc.

Similar sites:

1. CamelCamelCamel: It is a price tracking website that helps shoppers find the best deals on Amazon.
2. Honey: It is a browser extension that helps shoppers find the best deals by automatically applying coupon codes and price drops to their purchases.
3. RetailMeNot: It is a website and app that provides shoppers with coupons, promo codes, and discounts from a wide range of online retailers.
4. ShopSavvy: It is a barcode scanning app that allows shoppers to compare prices in-store and online to find the best deals.
5. SlickDeals: It is a community-driven website that posts the best deals from across the web.
6. MySmartPrice: It is a price comparison website and app that helps shoppers find the best deals on a wide range of products.
7. PriceBaba: It is a price comparison website and app that provides users with up-to-date pricing information and product specifications.
8. CompareRaja: It is a price comparison website that allows shoppers to compare prices from multiple online retailers in India.
9. PriceDekho: It is a price comparison website and app that provides users with detailed product information and price comparisons.

Proposed Model

Wishlist is an application completely built with python with a user-friendly interface that makes online shopping easier by showing you a side-by-side comparison of amazon and flipkart prices for any product you search. Product can be searched by entering by simply entering the name in the search-field and pressing a button. This will use web scraping to get the price and related information for you using beautifulsoup4 in python and display the results.

If you want to save the outputted information for future reference, you can add it to your *Wishlist* database by pressing another button on the side. The data will be saved in a MySQL database in two different tables for amazon and flipkart.

Similar to adding a product, it can be removed just as easily by entering its key and hitting confirm. This will use SQL queries to drop that tuple from both of the tables and update your list.

A side-by-side view makes for a better comparison and will make the decision-making process much faster and easier.

It has another feature 'Discover latest in TECH' that shows you articles of latest tech releases and keeps you updated. Aside from a great functionality addition to the shopping experience, is a tool to just spend time online reading about technology. This uses a python api to get news from multiples websites. This api lets you select the country and category of the articles you want. In the *Wishlist* application, the country is set to India and category is set to technology.

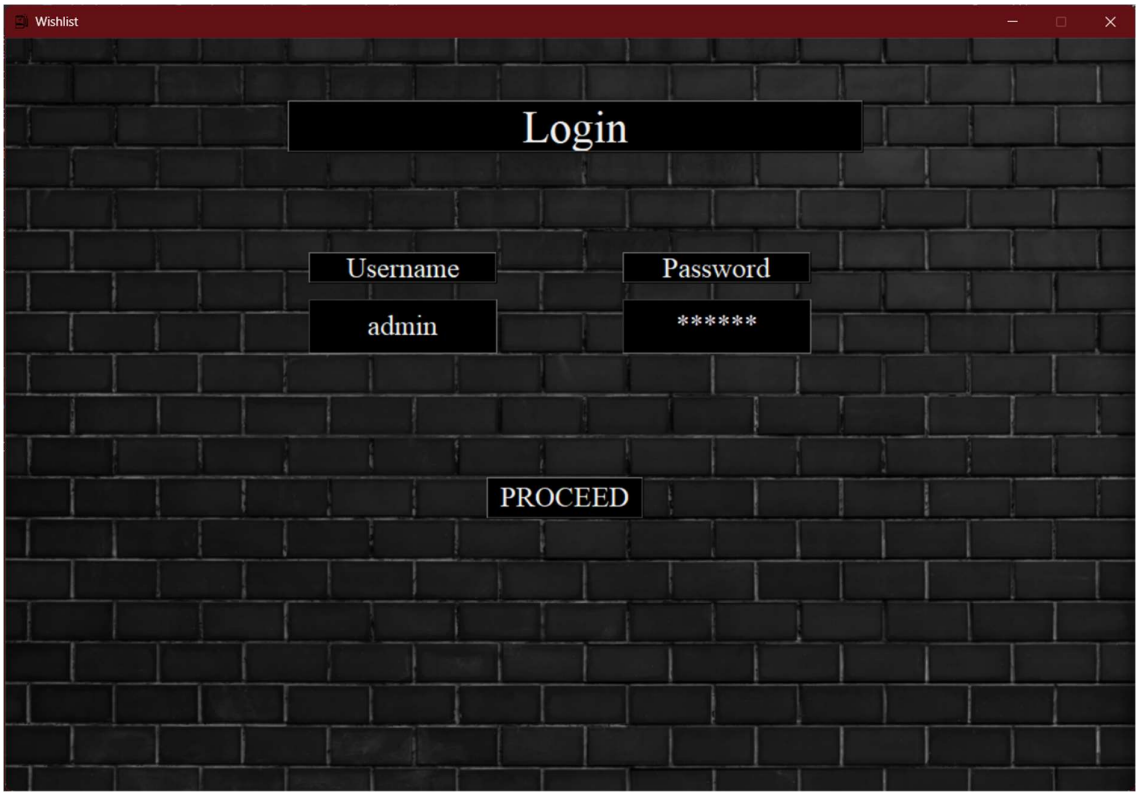
For all the products you search and even the tech articles, links are provided and can take you to the respective website in just a click of a button.

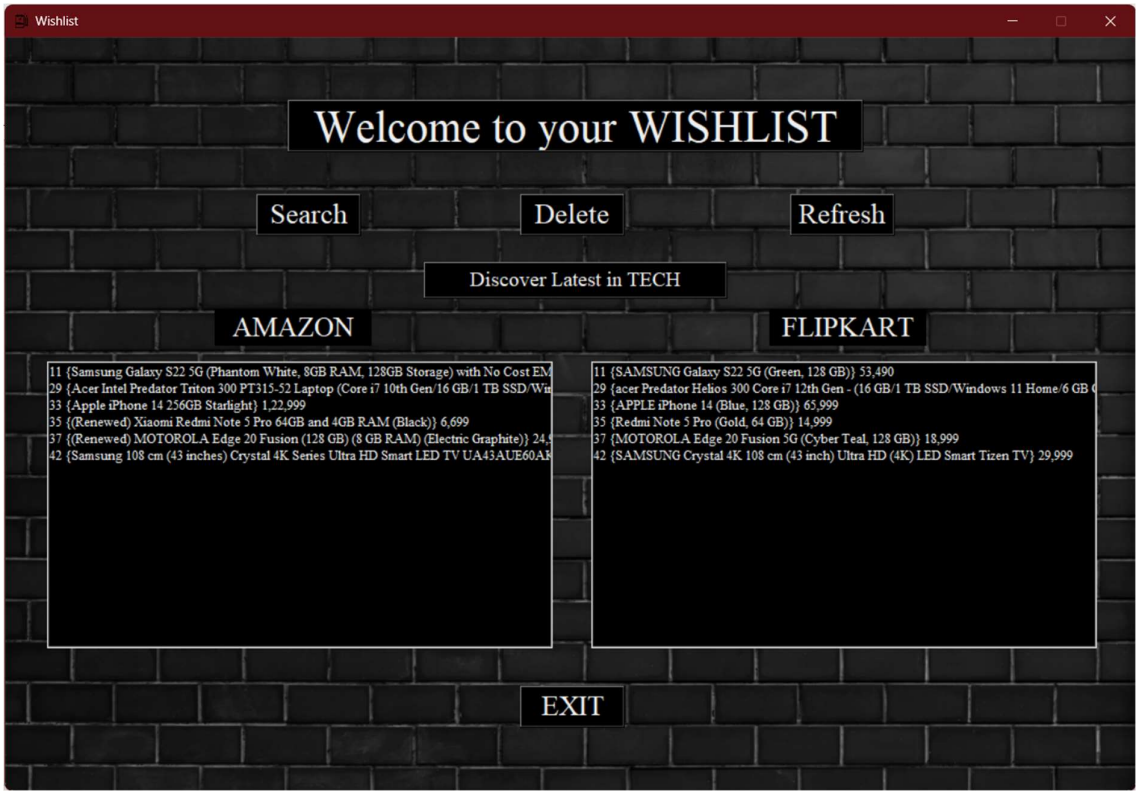
Buttons have been added to navigate between pages easily and a message is displayed after you perform an action, like deletion to avoid performing the action multiple times.

Experimental results

Login and homepage:

Login page uses a database to confirm legitimate user credentials and allows the user to use the services further. Upon entering incorrect password or username, it shows an error message, prompting the user to enter their details again.

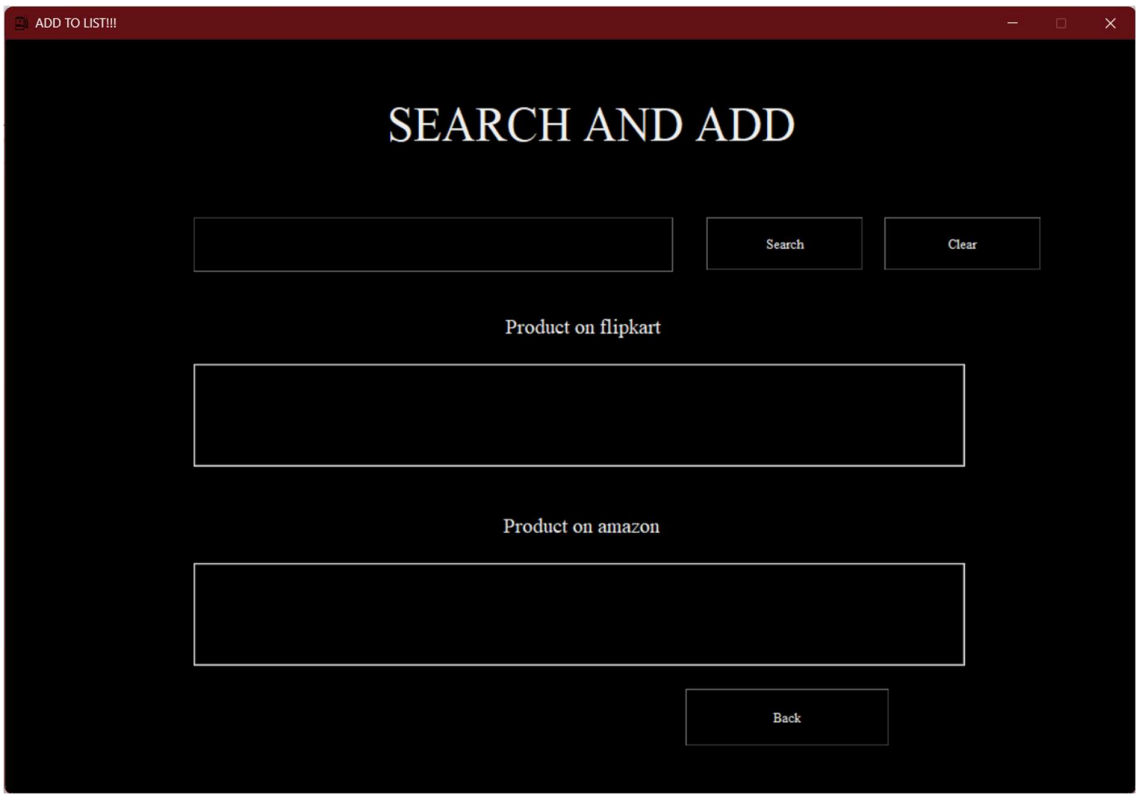




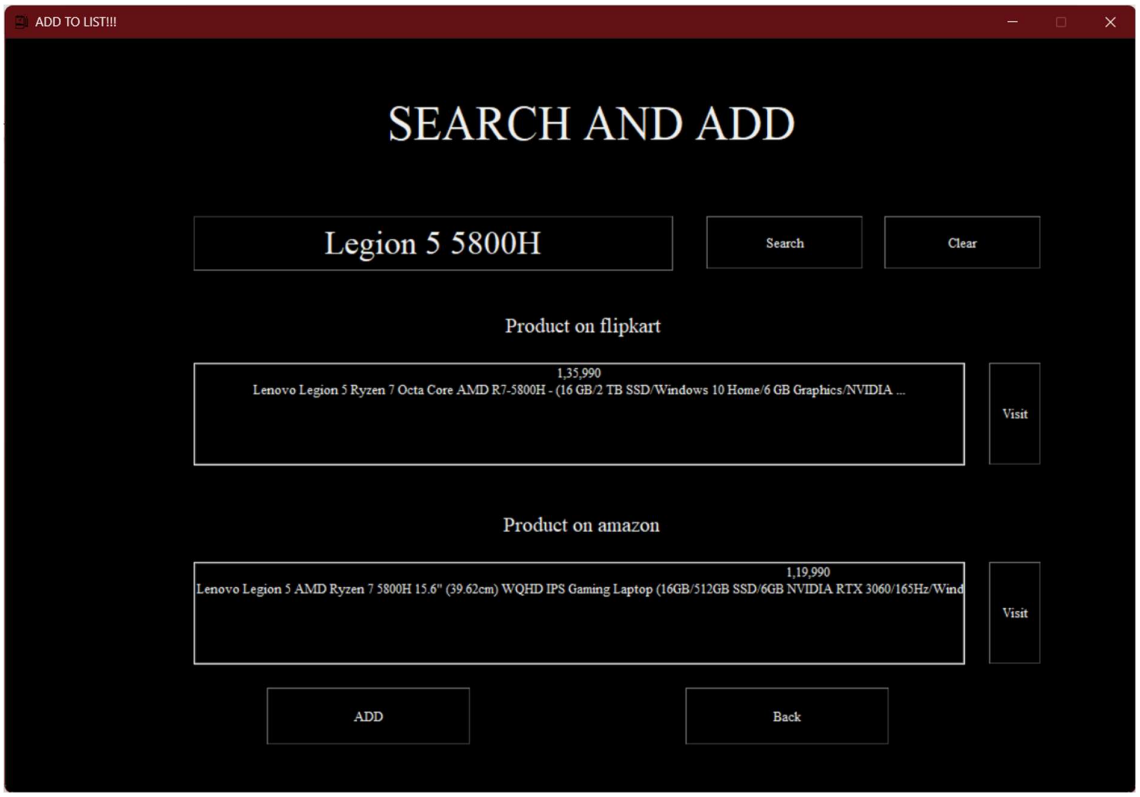
Search feature:

This uses web scraping via BeautifulSoup. Through experimentation search function was modified multiple times for maximum effectiveness. The search query is broken down using split function and then those individual terms are searched in the title tag of the respective website's parsed html code obtained through BeautifulSoup.

Search Window:



“Add” and “visit” buttons appear:

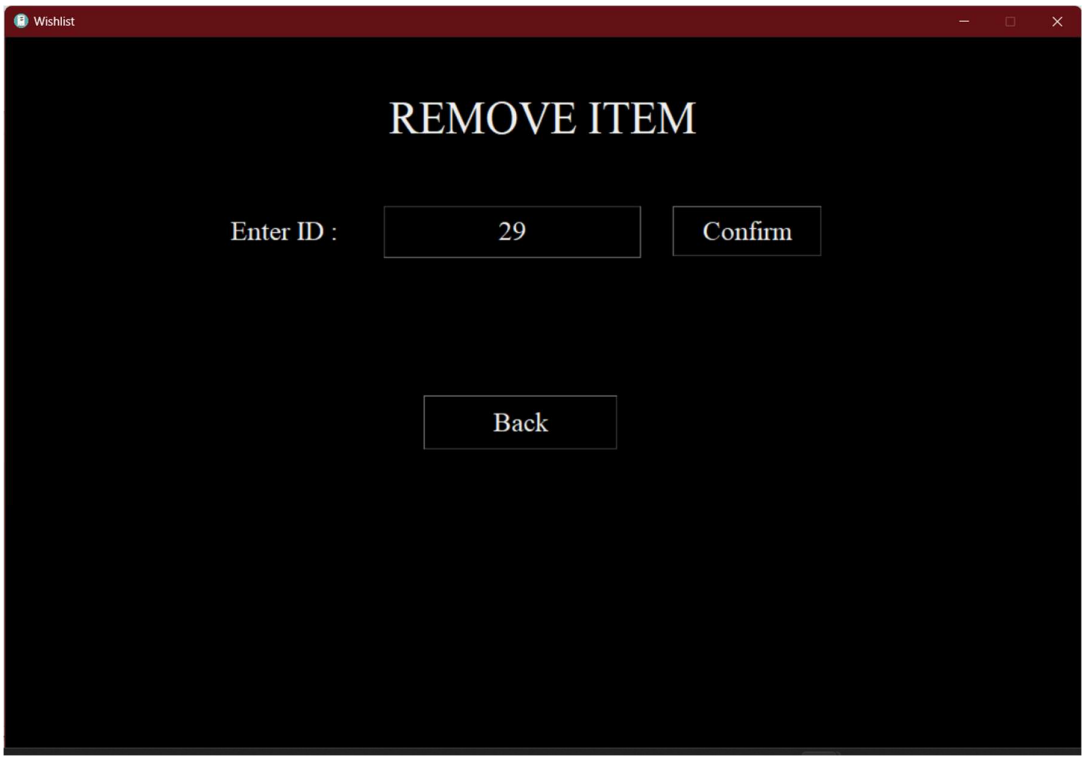


Deletion and Addition:

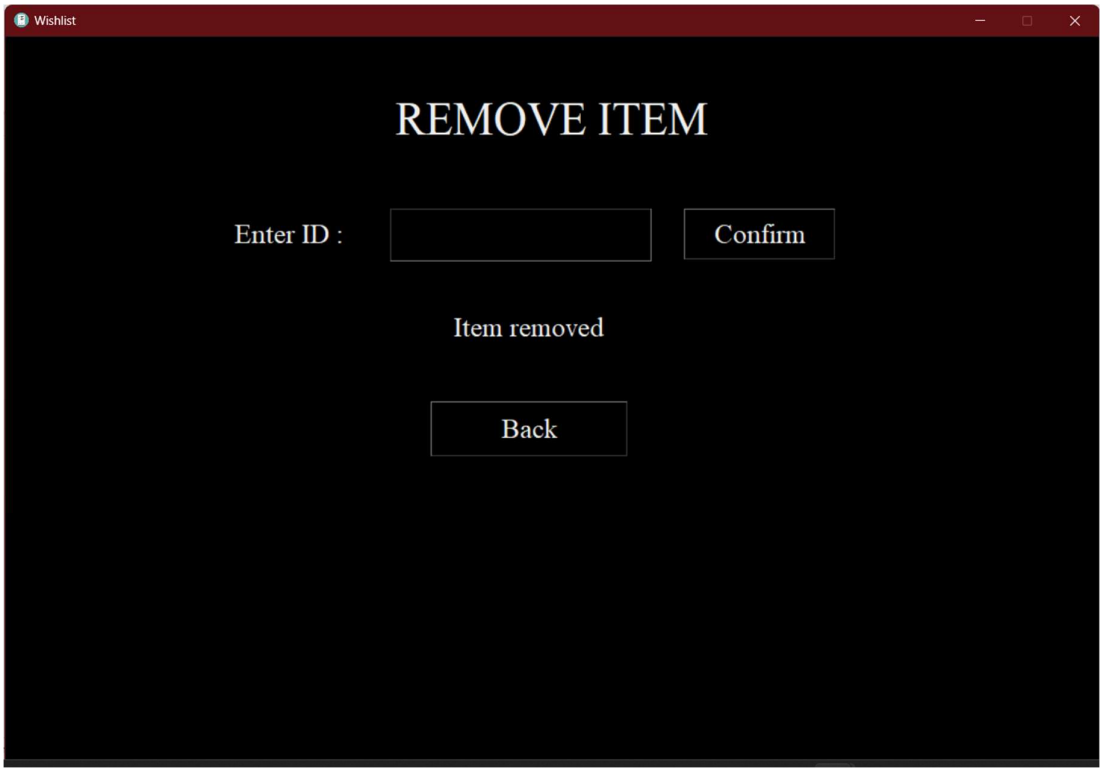
To add MySQL queries are used to update the table. Auto increment is added to serial no. of items so as to organise them and make deletion easier. After deletion serial no. of the other items is re-arranged to make the list numbering continuous.

Delete Window:

Deletion operation:

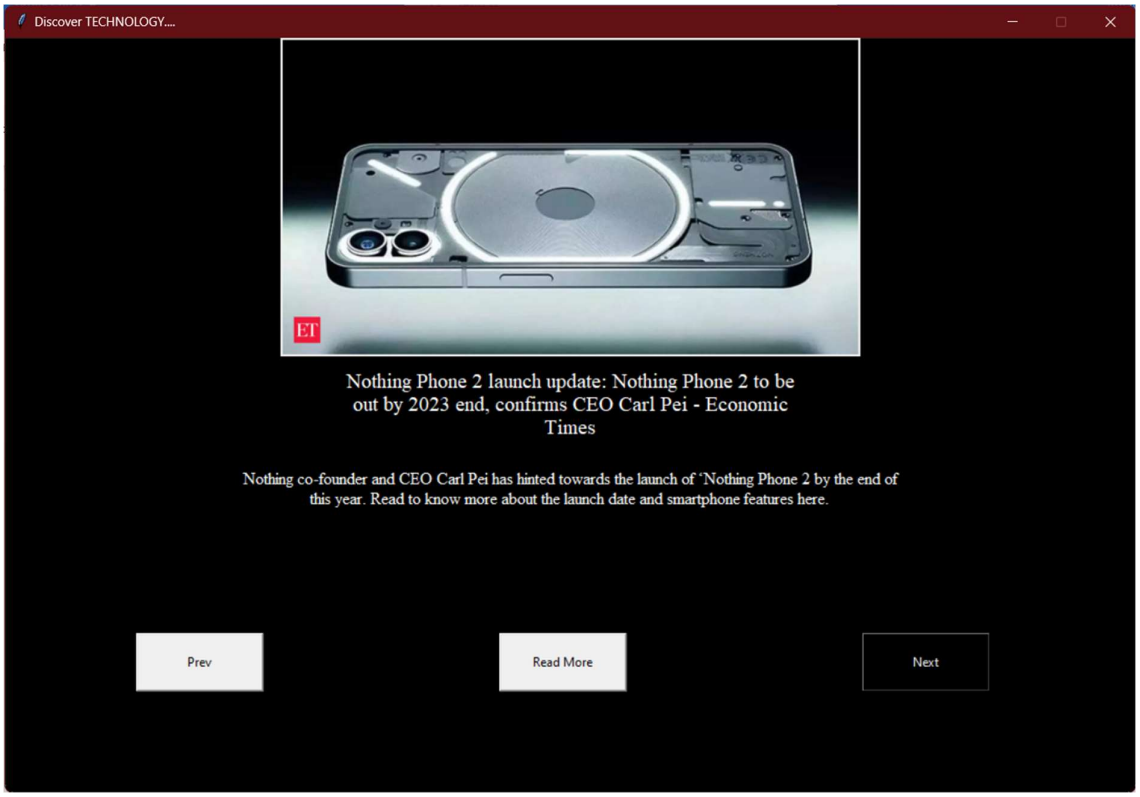


Deletion Confirmed:



Latest in TECH feature:

This uses a python api to get a json file of all the information you want. From this json file necessary data can be selected to be displayed like the image, title, description, etc.



Refresh:

This is used to refresh the list every time you add or delete an item. Data from the MySQL table is already updated using queries, this is just used by the tkiner to get the latest updated information and make necessary changes.

Web Scraping:

Through experimentation, it was decided to use BeautifulSoup for web-scraping, to its ease of use and fast results.

Several other modules were taken into consideration, but rejected after trials, selenium, autoscraper being among them. Autoscraper is just also quite effective, however still in development phase as it is quite new.

Tkinter:

Tkinter was chosen over Django and flask after experimentation due to its ease of use. This has a small learning curve and decent looking, working applications can be made even in the beginning stages of learning. Django and flask are also popular and widely used, however for a project of this size, tkinter was the obvious choice. In tkinter, grid, place and pack were used to align widgets in their respective windows.

Conclusion

The application was successfully completed and runs perfectly. Any item can be searched in the search bar, and a result is displayed if it is available on that specific e-commerce platform. Web-scraping has been done quite effectively as even detailed queries return a search result.

These results can then be added to the *Wishlist* database and retrieved again, as one pleases to be displayed to the user. On deleting an item, it disappears from the list after you click refresh and the database is updated accordingly.

The ‘Latest in TECH’ feature works like a wonder. With easy to navigate interface, beautiful, easy-on-the eyes UI it keeps you updated with the ever-evolving tech world and moreover keeps you updated.

In conclusion, this application can become a regularly used tool for any person due to its versatility and real-life application combined with its classic, old-school interface. Whether you're a casual or avid shopper, you can benefit from using *Wishlist* to make better purchasing decisions, save time and money, and enjoy a better shopping experience.