PriceHub

Final Report

CSC 59866 - M - Senior Design II

Prof. George Wolberg

December 11th, 2022

Contributors:

Carlos Flores
☎ (646) 726-2867
📧 cflores005@citymail.cuny.edu

Ahmad Adil
☎ (917) 474-8200
📧 aadil002@citymail.cuny.edu

Steven Granaturov
☎ (646) 785-9453
📧sgranat000@citymail.cuny.edu

Jia Cong Lin
☎ (929) 888-2903
📧 jlin023@citymail.cuny.edu

Table of Contents:

★ Introduction

Issue: People enjoy shopping at several locations. This results in common problems shoppers face, too many prices to keep track of! Nonetheless, your online shopping experience will be more enjoyable thanks to our app, PriceHub! Offering a wide range of products from well-known shop names like Amazon, Walmart, and eBay, our goal is to help you save your time and money.

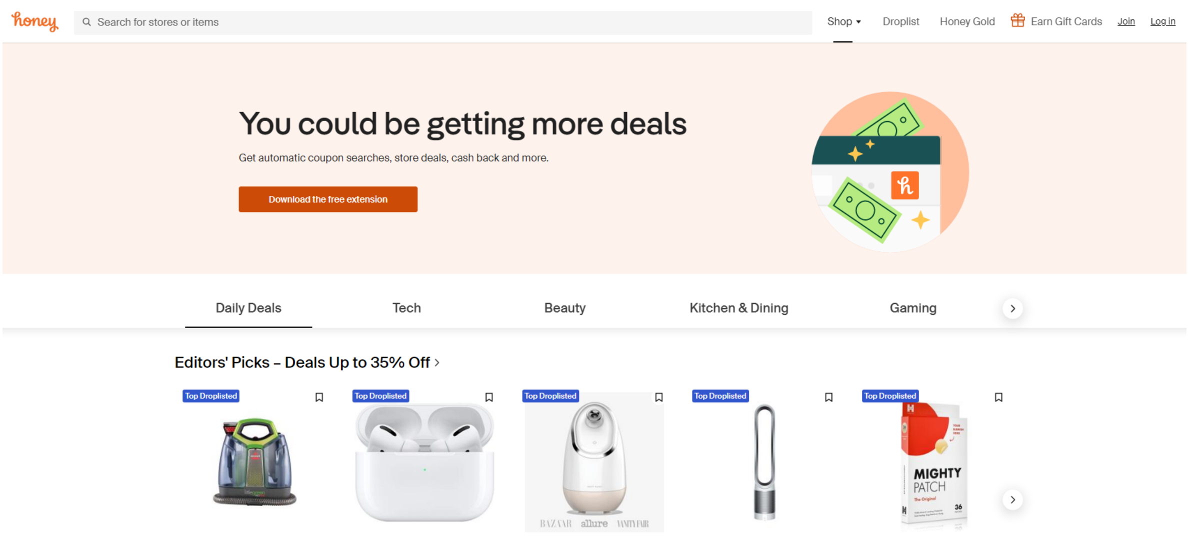
By tracking the best prices for the items of your choosing in our app, you're not only saving yourself the tedious process of looking for the item on countless different websites; but also saving yourself time that you can further utilize in other activities.

We value each of our users, hence we want to provide them with the best online shopping experience possible, helping them save money in the process. With this the target audience would be anyone who is looking for the best deals on their favorite products.

★ Previous Work

The idea of comparing prices in different stores is not something new. Nonetheless, it hasn't been exploited to its full potential. Currently, price comparisons between different stores are only available for selected sectors, mainly electronics and high-value rare items, such as sneakers. This same scenario applies to availability as stock trackers are very limited. To further emphasize this, some of the current competitors and their functionalities are as follows:

Honey: https://www.joinhoney.com/



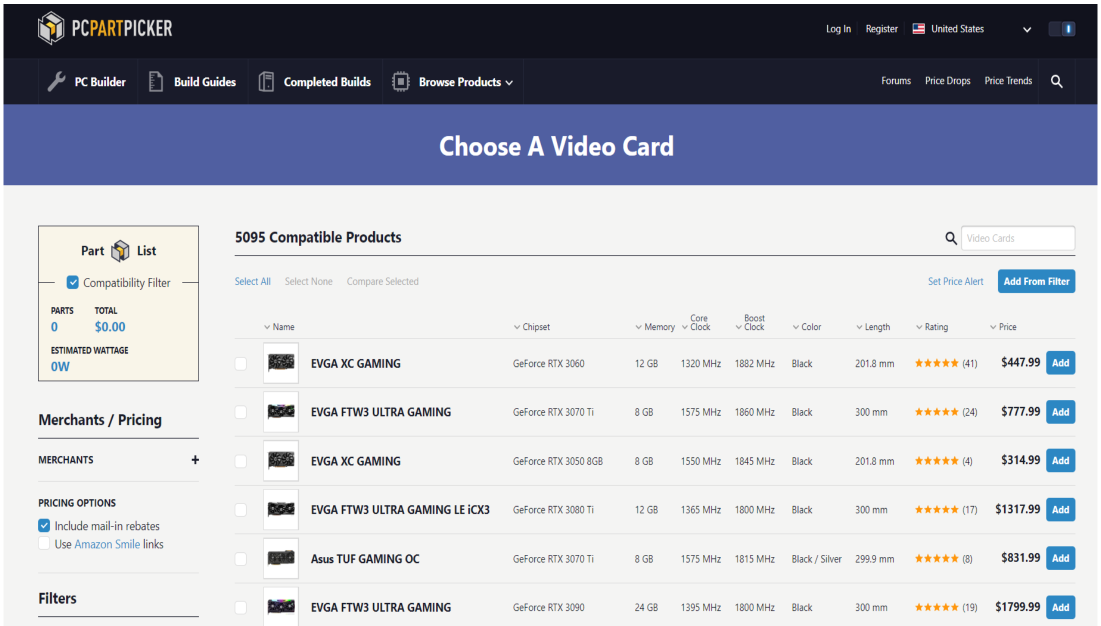
You could be getting more deals

Get automatic coupon searches, store deals, cash back and more.

Pros	Cons
Includes price history	Is not a stock tracker
Has discounts	Is not a mobile app
	Only shows the price history of the current store the user is on.

Honey is a browser extension that applies coupons and helps users save money by providing price history, determining if it is the best time to buy, and applying affiliated coupons at checkouts. This is great, but it is a letdown that no such app exists with these functionalities.

PcPartPicker: https://pcpartpicker.com/



Choose A Video Card

Part List

☑ Compatibility Filter

PARTS 0 TOTAL $0.00

ESTIMATED WATTAGE 0W

Merchants / Pricing

MERCHANTS +

PRICING OPTIONS

☑ Include mail-in rebates

□ Use Amazon Smile links

Filters

5095 Compatible Products

Video Cards

Select All Select None Compare Selected

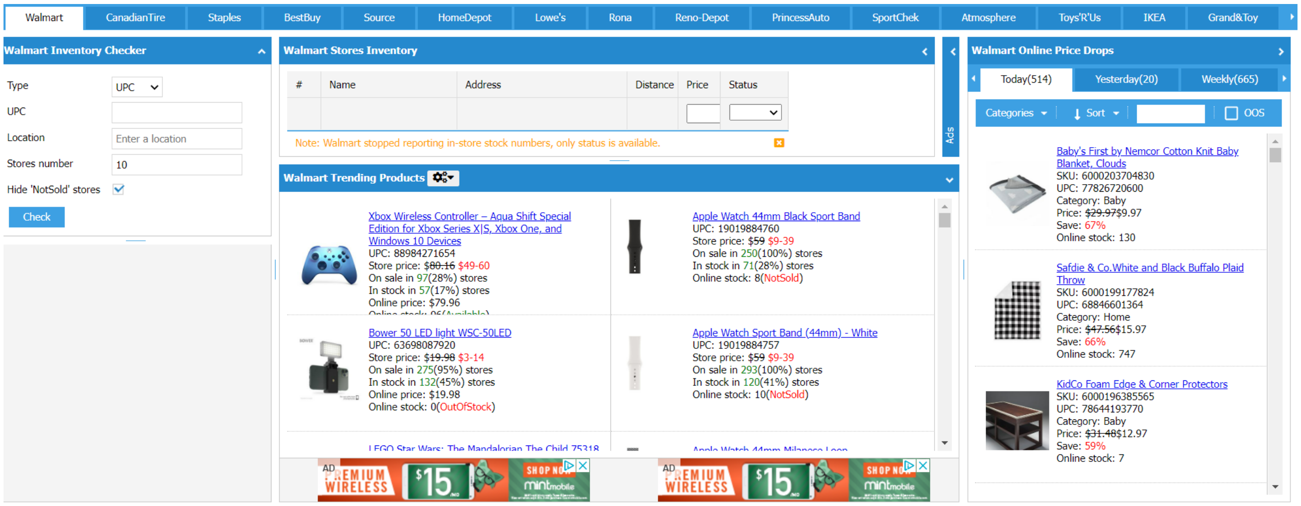
Set Price Alert Add From Filter

Name	Chipset	Memory	Core Clock	Boost Clock	Color	Length	Rating	Price	
EVGA XC GAMING	GeForce RTX 3060	12 GB	1320 MHz	1882 MHz	Black	201.8 mm	★★★★★ (41)	$447.99	Add
EVGA FTW3 ULTRA GAMING	GeForce RTX 3070 Ti	8 GB	1575 MHz	1860 MHz	Black	300 mm	★★★★★ (24)	$777.99	Add
EVGA XC GAMING	GeForce RTX 3050 8GB	8 GB	1550 MHz	1845 MHz	Black	201.8 mm	★★★★★ (4)	$314.99	Add
EVGA FTW3 ULTRA GAMING LE iCX3	GeForce RTX 3080 Ti	12 GB	1365 MHz	1800 MHz	Black	300 mm	★★★★★ (17)	$1317.99	Add
Asus TUF GAMING OC	GeForce RTX 3070 Ti	8 GB	1575 MHz	1815 MHz	Black / Silver	299.9 mm	★★★★★ (8)	$831.99	Add
EVGA FTW3 ULTRA GAMING	GeForce RTX 3090	24 GB	1395 MHz	1800 MHz	Black	300 mm	★★★★★ (19)	$1799.99	Add

Pros	**Cons**
Has a user-friendly UI, that allows for quick access	The site is dedicated to computer parts only, limiting its audience
Part selection for items is large and can search large retail stores, like Amazon, Walmart, Best Buy	No mobile app exists that can notify users for price drops since its a web site first design

PcPartPicker is a comparison online shopping website that focuses on Electronics, specifically computer components and peripherals. One of the interesting things that PcPartPicker excels at, is being able to get data from multiple sources and allowing users the ability to compare prices between the different retailers. Since PcPartPicker focuses on computer electronics their selection of stores is mostly limited to retailers like, Amazon, Newegg, BestBuy, B&H, Adorama and others. The downside with PcPartPicker, however, is that it is only offered on a website and not as an app.

StockTrack: https://stocktrack.ca/



Type: UPC
UPC:
Location: Enter a location
Stores number: 10
Hide 'NotSold' stores ☑

#	Name	Address	Distance	Price	Status

Note: Walmart stopped reporting in-store stock numbers, only status is available.

Xbox Wireless Controller – Aqua Shift Special Edition for Xbox Series X|S, Xbox One, and Windows 10 Devices
UPC: 88984271654
Store price: ~~$80.16~~ $49-60
On sale in 97(28%) stores
In stock in 57(17%) stores
Online price: $79.96
Online stock: 96(Available)

Apple Watch 44mm Black Sport Band
UPC: 19019884760
Store price: ~~$59~~ $9-39
On sale in 250(100%) stores
In stock in 71(28%) stores
Online stock: 8(NotSold)

Bower 50 LED light WSC-50LED
UPC: 63698087920
Store price: ~~$19.98~~ $3-14
On sale in 275(95%) stores
In stock in 132(45%) stores
Online price: $19.98
Online stock: 0(OutOfStock)

Apple Watch Sport Band (44mm) - White
UPC: 19019884757
Store price: ~~$59~~ $9-39
On sale in 293(100%) stores
In stock in 120(41%) stores
Online stock: 10(NotSold)

LEGO Star Wars: The Mandalorian The Child 75318

Apple Watch 44mm Milanese Loop

Baby's First by Nemcor Cotton Knit Baby Blanket, Clouds
SKU: 6000203704830
UPC: 77826720600
Category: Baby
Price: ~~$29.97~~$9.97
Save: 67%
Online stock: 130

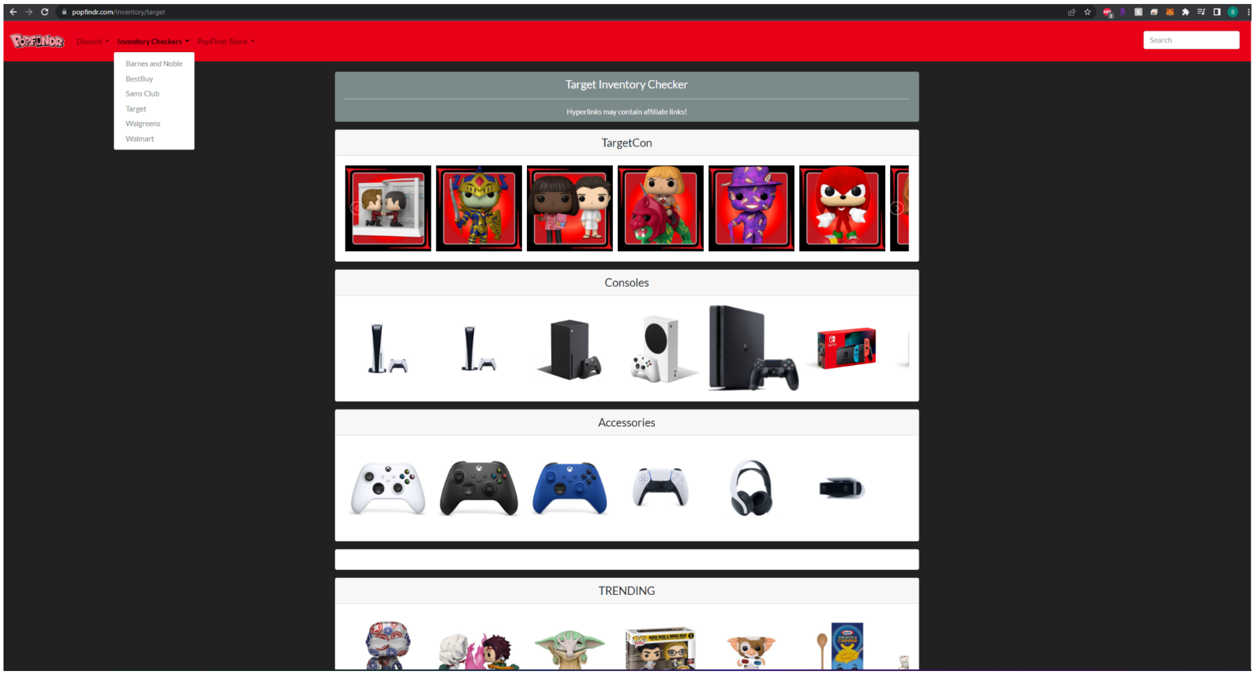
Safdie & Co.White and Black Buffalo Plaid Throw
SKU: 6000199177824
UPC: 68846601364
Category: Home
Price: ~~$47.56~~$15.97
Save: 66%
Online stock: 747

KidCo Foam Edge & Corner Protectors
SKU: 6000196385565
UPC: 78644193770
Category: Baby
Price: ~~$31.48~~$12.97
Save: 59%
Online stock: 7

Pros	**Cons**
Ability to search through a big selection of stores	Terrible User-Interface
Displays search results quickly	Searches product through the selected store, and doesn't show prices on other stores until that store is selected.
Accessible on any browser	Not a mobile app

StockTrack is a website featuring a variety of website tabs at the top for the user to search their products through. This is not as efficient as it could be as it forces the user to search the product individually, creating more work. The website provides links which will redirect you to another tab containing the product you want to look at. The way to navigate through and browse products on the website is terrible for the user due to their complexity.

PopFindr: https://popfindr.com/



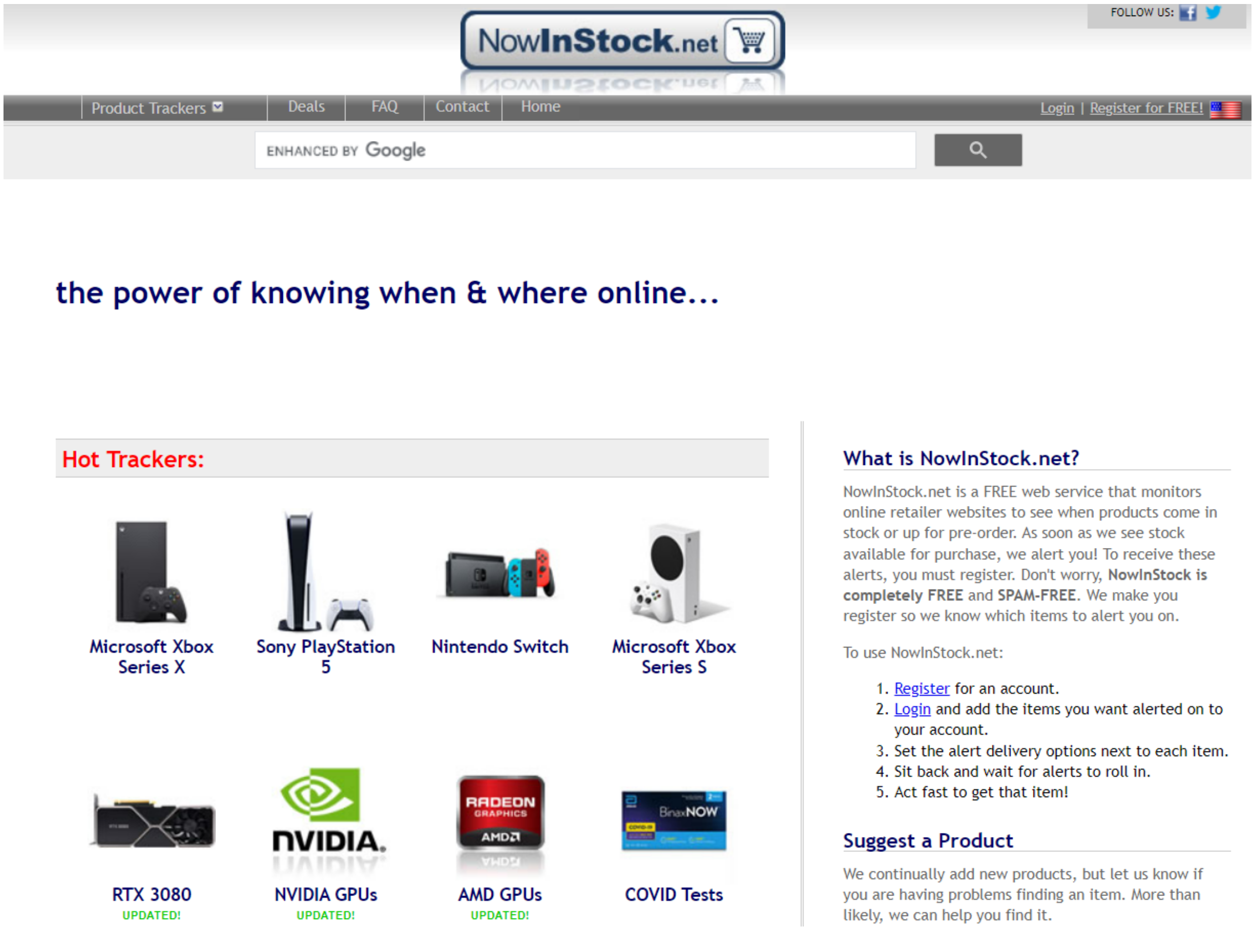
Target Inventory Checker

Hyperlinks may contain affiliate links!

Pros	**Cons**
Variety of API	Limited product shown
Ability to search through different stores	Only able to search through one store at a time
Simple User-Interface	Lack of features - No price history

PopFindr is a website originally for pops which are figurines of popular characters. It then turned into a site that is targeting gamers. The products it features are limited, but with a variety of API and simple user interface which appeals to the eye. PopFindr also lacks the features of price comparison between stores.

NowInStock: https://www.nowinstock.net/



the power of knowing when & where online...

What is NowInStock.net?

NowInStock.net is a FREE web service that monitors online retailer websites to see when products come in stock or up for pre-order. As soon as we see stock available for purchase, we alert you! To receive these alerts, you must register. Don't worry, **NowInStock is completely FREE** and **SPAM-FREE**. We make you register so we know which items to alert you on.

To use NowInStock.net:

1. Register for an account.
2. Login and add the items you want alerted on to your account.
3. Set the alert delivery options next to each item.
4. Sit back and wait for alerts to roll in.
5. Act fast to get that item!

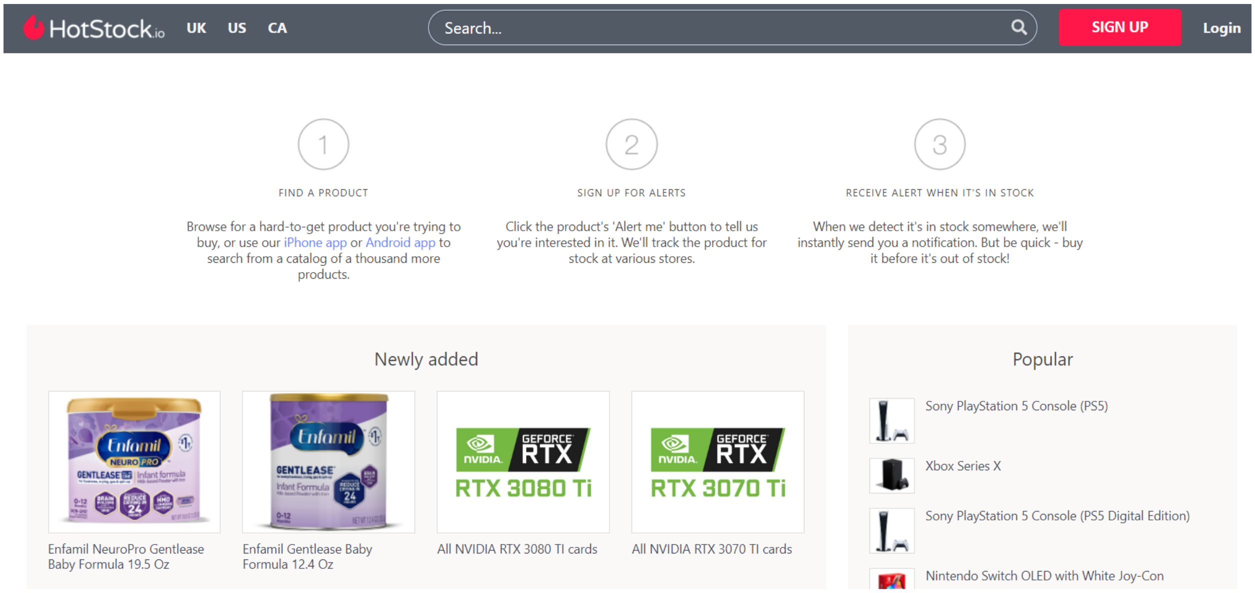
Suggest a Product

We continually add new products, but let us know if you are having problems finding an item. More than likely, we can help you find it.

Pros	**Cons**
Lots of freedom in showcasing products	Not very user-friendly
Infinite library as it uses the Google Search Engine API.	When searching for a product, the organization of the retailers, price, availability and price history is not appealing, and would cause users difficulty with navigation.

NowInStock is also another website featuring a lot of items targeting gamers, but there is also a link where you could suggest a product for them to track. NowInStock has the idea of freedom in showcasing stocks and an infinite library as it uses the Google Search Engine API. The interface is not very appealing and does not provide much help in navigating their website or providing a mobile version of the site. NowInStock also has an alarm for the user as a notification when something is in stock, but the organization is bad when listing the prices and availability of stocks.

HotStock: https://www.hotstock.io/



1
FIND A PRODUCT
Browse for a hard-to-get product you're trying to buy, or use our iPhone app or Android app to search from a catalog of a thousand more products.

2
SIGN UP FOR ALERTS
Click the product's 'Alert me' button to tell us you're interested in it. We'll track the product for stock at various stores.

3
RECEIVE ALERT WHEN IT'S IN STOCK
When we detect it's in stock somewhere, we'll instantly send you a notification. But be quick - buy it before it's out of stock!

Newly added

- Enfamil NeuroPro Gentlease Baby Formula 19.5 Oz
- Enfamil Gentlease Baby Formula 12.4 Oz
- All NVIDIA RTX 3080 TI cards
- All NVIDIA RTX 3070 TI cards

Popular

- Sony PlayStation 5 Console (PS5)
- Xbox Series X
- Sony PlayStation 5 Console (PS5 Digital Edition)
- Nintendo Switch OLED with White Joy-Con

Pros	Cons
Has a mobile app and alerts	No Prices or price comparisons between the different retailers
Allows users to show trending items that high demand from various categories	The search within the site is extremely slow and isn't able to browse the inventory of certain stores.

HotStock is a website with international availability and mobility. The price comparison between different retailers is also nonexistent and the searches are extremely slow with some errors in browsing certain sites. However, the interface is friendly and gives directions along with the trending items with high demand from various categories.

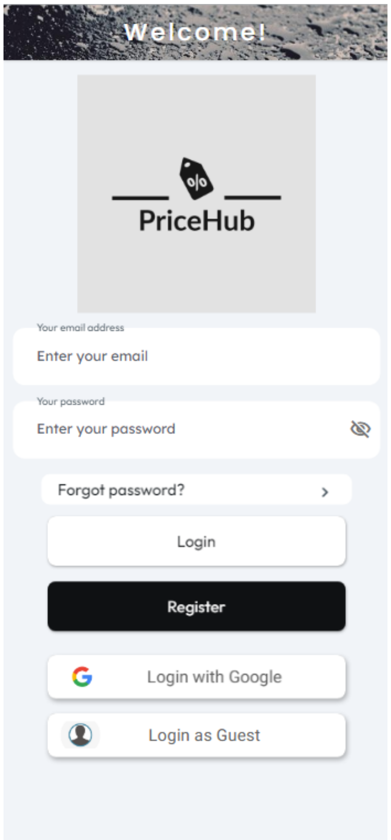
- ★ **Approach**

As observed by the previous point, there is no competitor that has a complete, functional application that provides users with all the facilities we seek to provide. Our goal is to grasp all the good qualities of competing software and provide an upgrade where possible. Combining what's already working in one place guarantees success in the app's popularity and usefulness. We want users to feel satisfied with our service as we are aiming to provide them with a smooth, fast and efficient shopping experience.

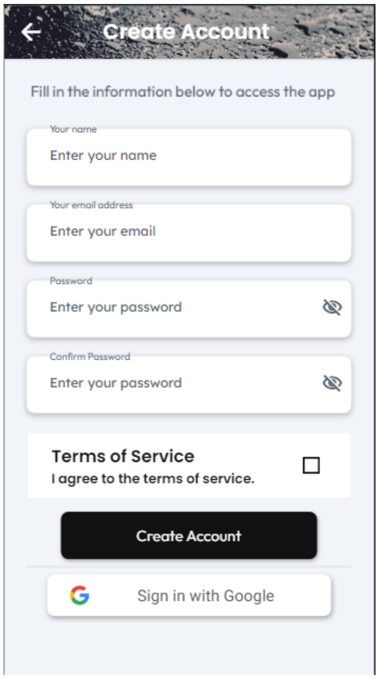
With this in mind, the building blocks of our app are as follows:

1. Log in as a guest or personal account: offers you the following perks
   a. Language
   b. Profile customizations
   c. Display Name
   d. Feedback
2. Best Deals: provides you with the best deals of the day
3. Categories: different sets of selected categories where to choose from
4. Product Display: a dynamic page where items are being displayed when selected
   Allows you to:
   a. Compare Prices
   b. Favorite items
5. Redirect to Store Page: brings you to the official stores to purchase the goods securely.

- Next, the user interface was divided into six main parts:
  - Login page
    - The login page allows users to login using email or Google Authentication. Or alternatively, as a guest.



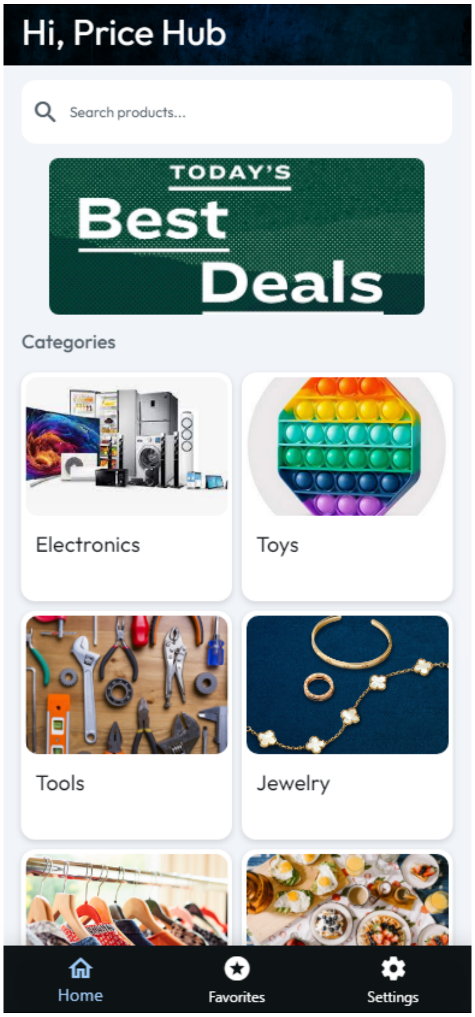
- Sign-up page
  - The sign-up page allows users to sign up utilizing email or Google Authentication.


Fill in the information below to access the app

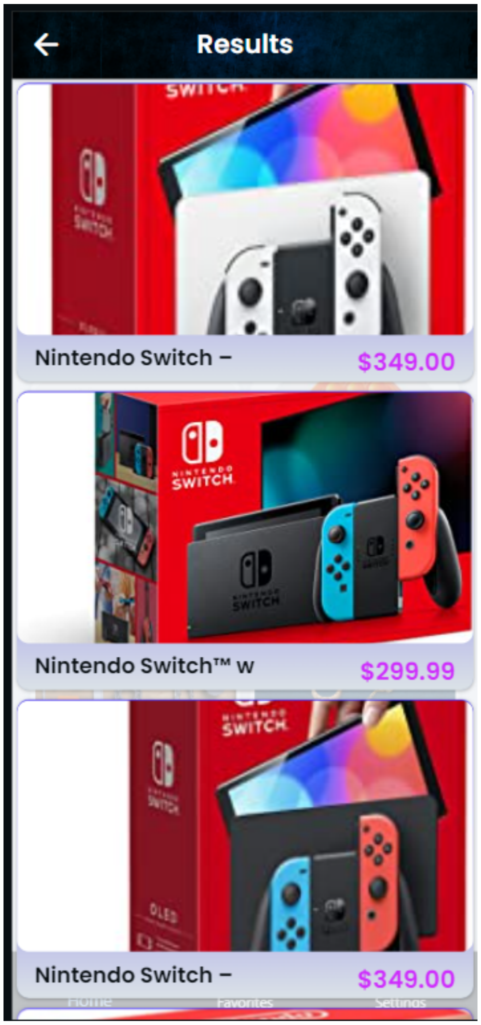
Terms of Service

I agree to the terms of service.

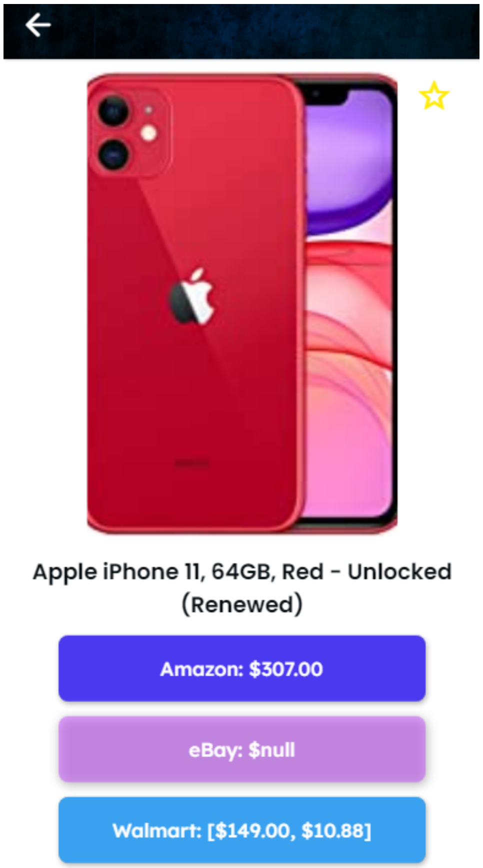
- Categories/Search
    - This would allow users to quickly see deals for products in their respective categories.
    - Search is the main focal point. Allows users to search for products and compare prices from various different stores.
    - Categories serve the role of the homepage after user login.
    - Users can then proceed to search for an item of their choosing or visit the best deals of the day or a preferred category.
        - Sends the requested product or category to Amazon through the API.



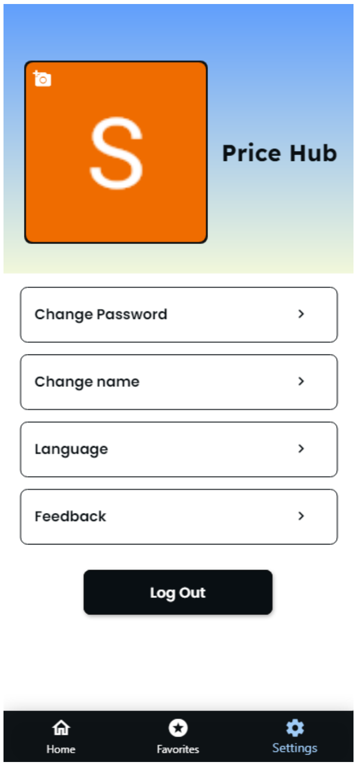
- Results
    - A comprehensive results page that allows for products to be dynamically displayed
    - The results page appears from the user's search, categories or best deals
    - The results page shows the list of products, with the respective Amazon price, from the user-initiated search.
    - Each Product is displayed automatically using an API call from the backend
    - The API returns a JSON file that contains crucial information such as product title, price, and image.



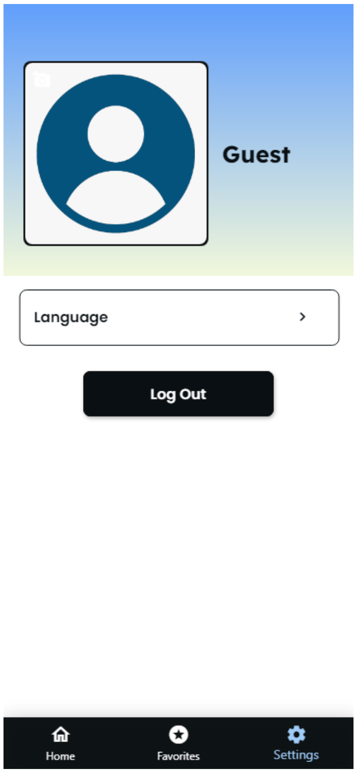
- Product Display
    - The product display page would display the product images, prices, and buttons to redirect users to the respective stores
    - Page consists of variables passed from the results page and presents the image, name, price, and URL to the user
    - The page pulls the name and image from Amazon
    - The page shows prices on Amazon, eBay, and Walmart
    - If the user wants to go to a specific store, the button will redirect them to the respective stores' website, leading them to the specific item search
    - Users can click the star icon at the top right to favorite the item


←
☆
Apple iPhone 11, 64GB, Red - Unlocked (Renewed)
Amazon: $307.00
eBay: $null
Walmart: [$149.00, $10.88]

- User profile page
    - This page allows registered users to take advantage of some of the app perks, which range from name and image customization to language, feedback, and password resetting.



- For guests, this view is more limited and defaulted, but still provides access to basic needs.

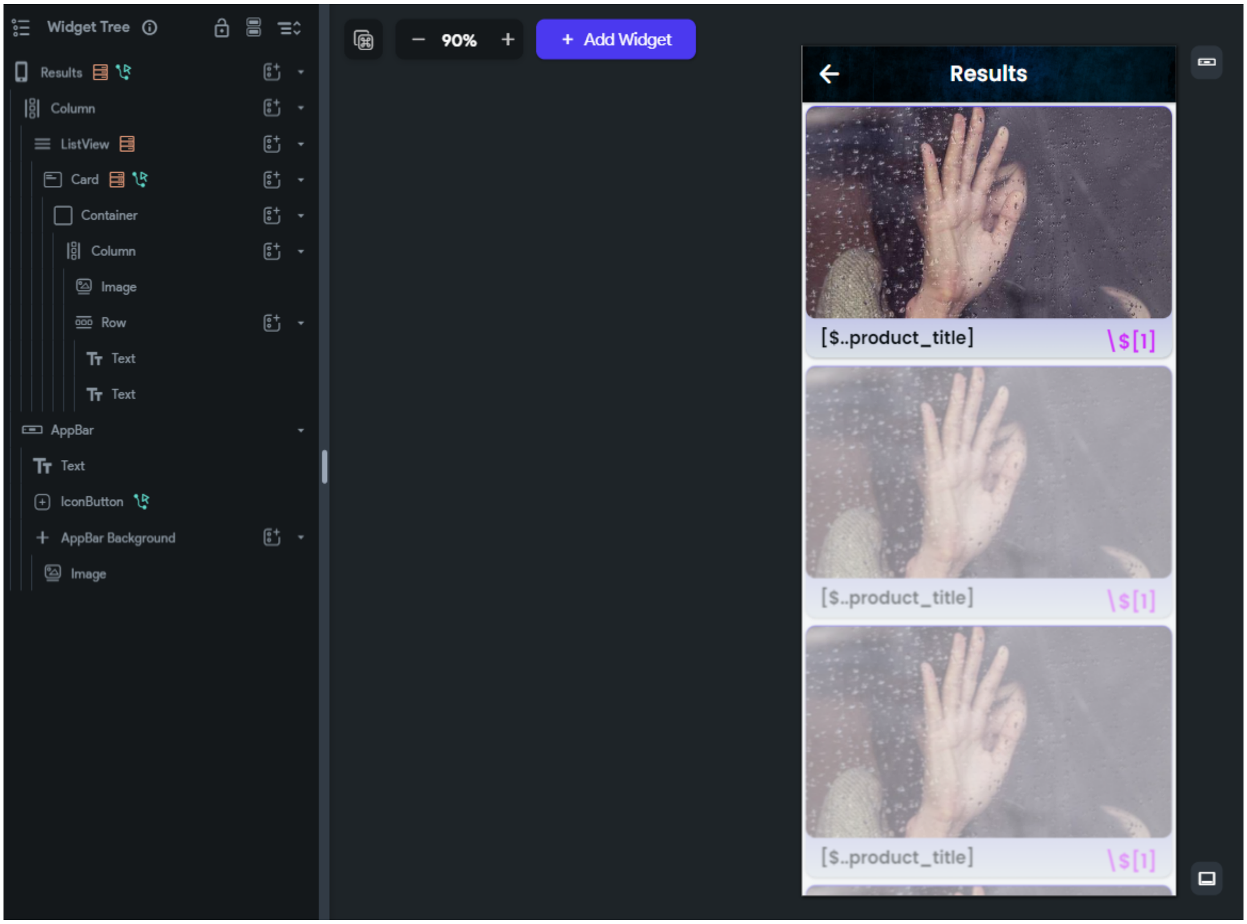


★ Implementation

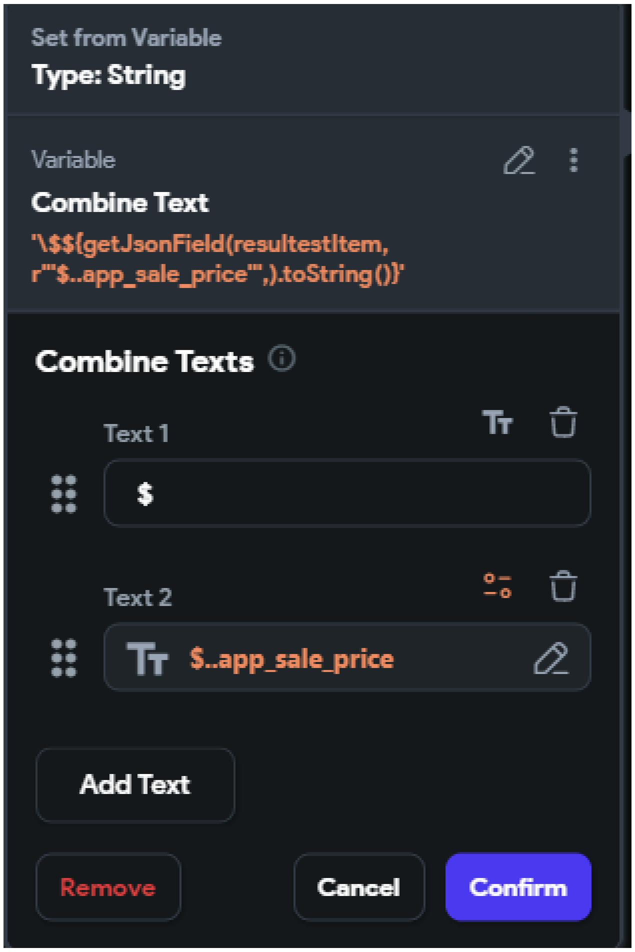
For the implementation of the app, we utilized a combination of Flutter, FlutterFlow, and respective store APIs to obtain product information and their prices.

- Libraries used:
  - FlutterFlow’s built in libraries for managing the user interface.
  - API’s that were sourced from RapidAPI with detailed documentation.
  - The eBay developer program allowed us to use the official eBay API as well as provided us with the eBay sandbox environment to play around with how the API works and configure it correctly to be ported over to FlutterFlow.
- Implementation solutions:
  - Most of the source code for our project consists mainly of FF (flutterflow) integrations through the use of the widget tree and the different components.

For instance, let’s dive into a specific section, and observe the following snippet:



This image shows the working environment for the results page, a major page in our application. To the left, we have the widget tree, which shows the structure of the components. The orange boxes are indicators of API calls, which provide us with the information that we are displaying to the user everytime they make a search for a specific item. The listview component holds the API call to amazon, which is the one that is first executed when we access this page. This API call provides us with a JSON file that we parse through JSON paths in order to display the data to the screen. You can see this present to the right of the image where the developmental app is running. These paths are present here in the name of the product ($..product_title) as well as the price ($[1]). Note that in this case price has the path $[1] because it is a composite path consisting of two strings being:



Set from Variable

Type: String

Variable

Combine Text

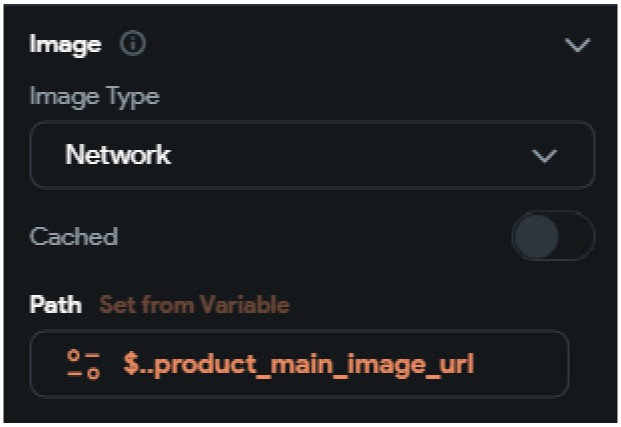
```
'\$${getJsonField(resultestItem,
r'''$..app_sale_price''',).toString()}'
```

Combine Texts

Text 1: $

Text 2: $..app_sale_price

The same principle applies to the image as the path is also a JSON path, just not visible from the experimental view



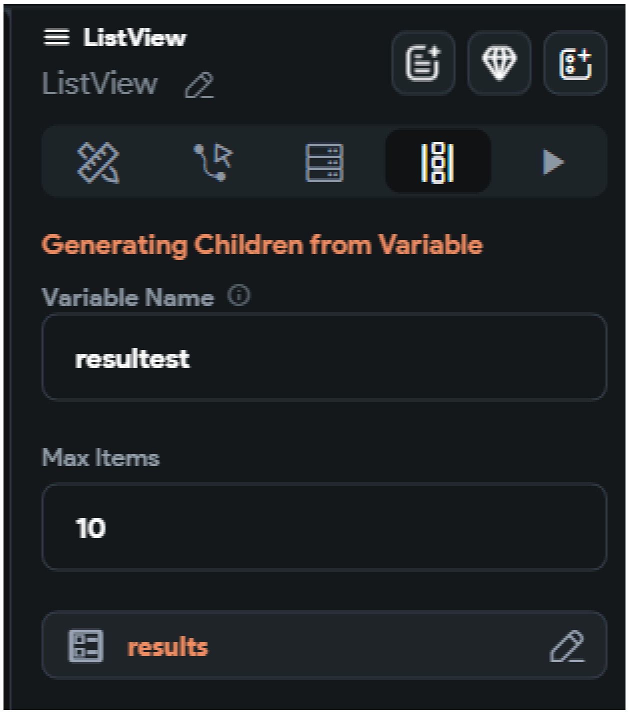
Image

Image Type: Network

Cached

Path Set from Variable: $..product_main_image_url

Next, I want to provide a further emphasis on the listview, but this time focusing on the dynamic children aspect of it. This is present as such:


ListView

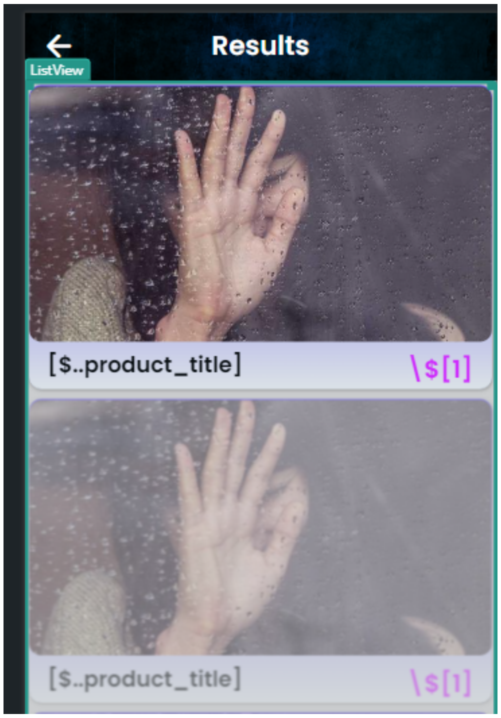
ListView

Generating Children from Variable

Variable Name: resultest

Max Items: 10

results

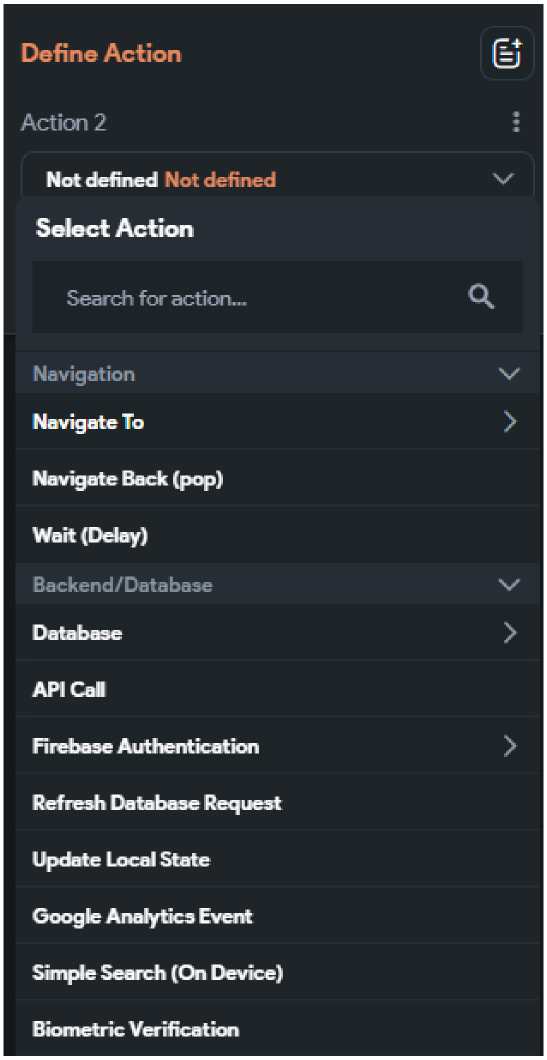


Dynamic children is basically the repeating pattern of the card that you see in the development area. This dynamic integration is what allows multiple items to be displayed on the app when we perform the API call to the Amazon website.

By performing the call, we are retrieving a JSON file with all the results, and we are populating the card dynamically, repeating the card over and over, and filling the specified paths as per the screenshots above.

Last, but not least important, I want to provide an emphasis on actions.

Actions are the green arrows you see in the working environment (refer to the card in the example above). Through actions, we can perform a variety of, well, actions. Following is a list of possible ones:



Define Action

Action 2

Not defined Not defined

Select Action

Search for action...

Navigation

Navigate To

Navigate Back (pop)

Wait (Delay)

Backend/Database

Database

API Call

Firebase Authentication

Refresh Database Request

Update Local State

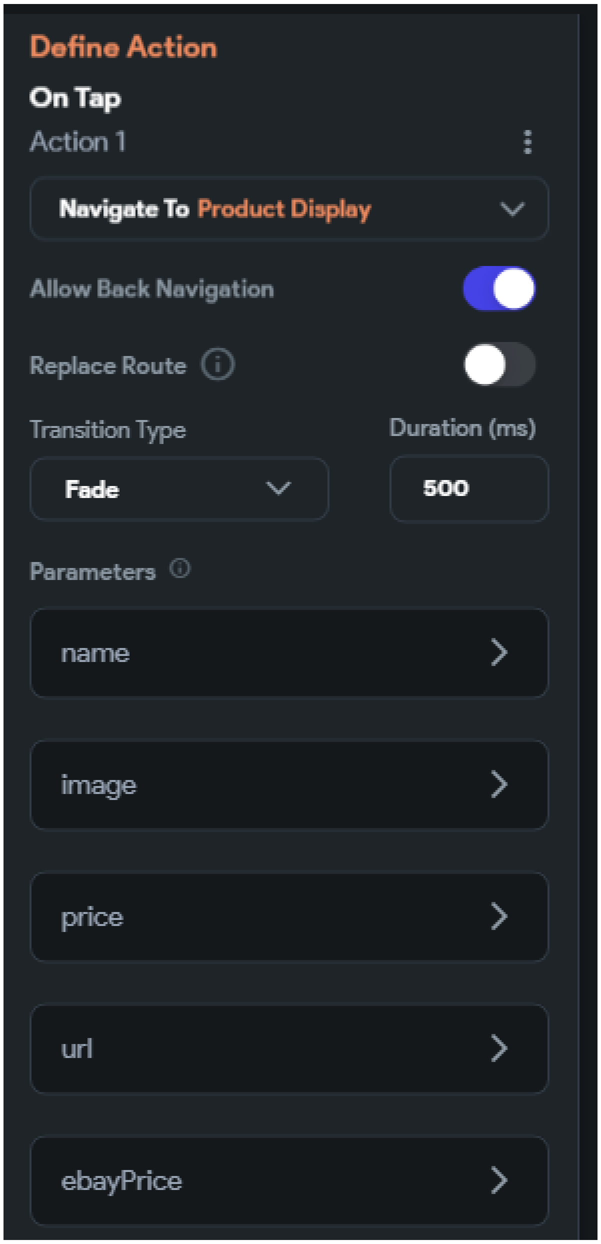
Google Analytics Event

Simple Search (On Device)

Biometric Verification

And many more!

In this case, specifically, the action present in the card allows us to transition to the product display page once an item is selected so that the magic occurs and the price comparisons can be made. Here is a closer look:



Define Action

On Tap

Action 1

Navigate To Product Display

Allow Back Navigation ●

Replace Route ○

Transition Type: Fade

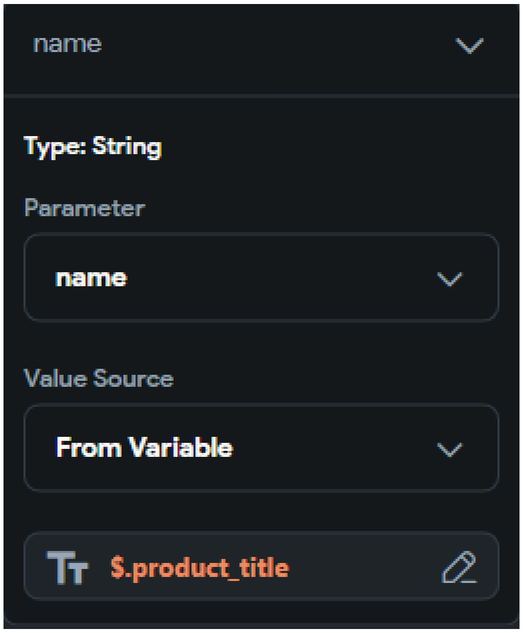
Duration (ms): 500

Parameters

- name
- image
- price
- url
- ebayPrice

Basically, as we are transitioning to product display, we are also passing parameters, which are fed with information by the specific API calls in the form of JSON paths.

For example:



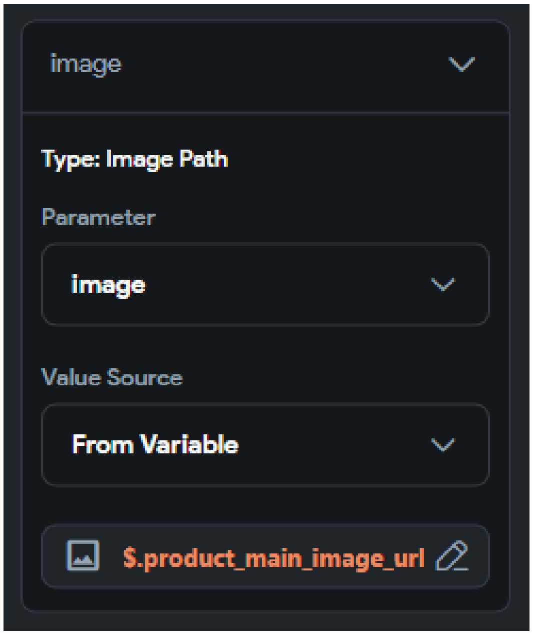
name

Type: String

Parameter: name

Value Source: From Variable

$.product_title



image

Type: Image Path

Parameter: image

Value Source: From Variable

$.product_main_image_url

Hence, this is how we were able to populate and manage data transferring between the application and API calls.

A similar approach is also taken for all the other attributes of the app. All of the code consists of components and specific actions that are performed under certain circumstances or when clicked.

- Challenges and overcomings:
    - We faced a lot of API and UI challenges.

It was tough to figure out how to connect the API's to Flutterflow. After successfully connecting the API with Flutterflow, we had a hard time figuring out how to parse through the JSON files and implement the API calls in our app to show the relevant information to the user. We also had a problem with the eBay API where we had to renew the eBay token every few days to have access to their API. We also had UI problems where names of users were not shown or the list view was showing the same product instead of going through and showing the list of all the different available products.

We overcome these challenges by coming together as a group to research and provide insight on how to fix the problem while using Google and Youtube to search for possible solutions or tutorials. The FlutterFlow and eBay documentation was also a great help when we faced trouble with understanding how to work with FlutterFlow and eBay API. Lastly, we also asked for assistance from other groups if there was an issue we could not find the solution to online.

- Tools used

The tools we used for our PriceHub app are:

- FlutterFlow
  - We were able to design a front-end system and implement the back end to make the application functional.
  - We used the Pro plan in order to access more than 2 API calls, along with other features such as the implementation of using different languages.
- RapidAPI
  - A hub of public and private APIs created to obtain certain data and display on an app.
  - This is where we were able to obtain all of our APIs along with their documentation.
  - We pay RapidAPI to get a certain amount of monthly calls as the free plan doesn’t give us a sufficient amount of calls.
- eBay Developers Program
  - We used the official eBay API by signing up for their developer program.
  - The Ebay API was first tested within Ebay’s sandbox environment to

ensure it would and then be ported over to flutterflow.

- ➢ Discord
  - ○ Allow us for better communication and able to work on one page together when FlutterFlow allows only one person to work on a page at a time

★ **Division of labor**

Regarding the division of work, our plan was to divide the work into sections. Because the development phase is going to consist mainly of FlutterFlow designing and API connections, we divided the team based on their strengths. Two people took care of the app design, components and aesthetics, meanwhile the other two members took charge of retrieving information from the different APIs.

Carlos Flores	Steven Granaturov	Ahmad Adil	Jia Cong Lin
● Created UI designs ● Implemented API and UI connections ● Implemented Languages, Results, Product Display and Favorites	● Populated all the category pages and assisted on implementing other pages ● Researched numerous APIs and helped with implementations ● Took care of all the finances of the project	● Extensive API research and implementation ● API maintenance and changes. ● Tested and built the Ebay API and authentication	● Parsed through JSON of all the API implementations ● Helped with implementing the API's ● Fixed Bugs like Google Sign-in and words becoming invisible

★ Future improvements

We plan to focus more on adding more functionality, redesign aesthetics, speeding up the app, and incorporating revenue to further improve our app. In terms of functionality, we are looking to add more stores and more products through our API implementations. We also plan to improve upon the amount of features that will be presented to users and the amount of login options, such as Facebook or AppleID, for better user convenience. We can also redesign the app to make it more user friendly including making the time for API calls to be faster which will speed up the app and improve user convenience.

★ What we learned

From the project, we learned how to use FlutterFlow in terms of UI design and API implementations for both the back and front end of the project. We also learned how to navigate and connect Firebase to let our users make an account for our app or let them log in and connect through their Gmail account. We also learned a lot about the different types of API's from RapidAPI and the eBay Developer Program. We learned how to connect their API and how to navigate them to get what we want.

★ Conclusion

We have fully implemented Amazon, Walmart, and eBay into our application. Down the life cycle of this application, we would like to add support for more stores such as BestBuy, Target, Home Depot, and more. This will require us to find or create API's for these stores if not readily available. Furthermore, we would like to redesign the User Interface to be more aesthetically pleasing, provide more product information, and make good use of the screen space.

Moreover, we would like to implement ads based on user searches to provide us with some revenue, and to provide useful advertisements to the user. A great place to put those ads would be when the application is loading the search results as it does take some time at the moment. Lastly, another source of improvement is to increase the speed of the application. There will be users that won’t end up liking our product as it is too slow to fetch the results.