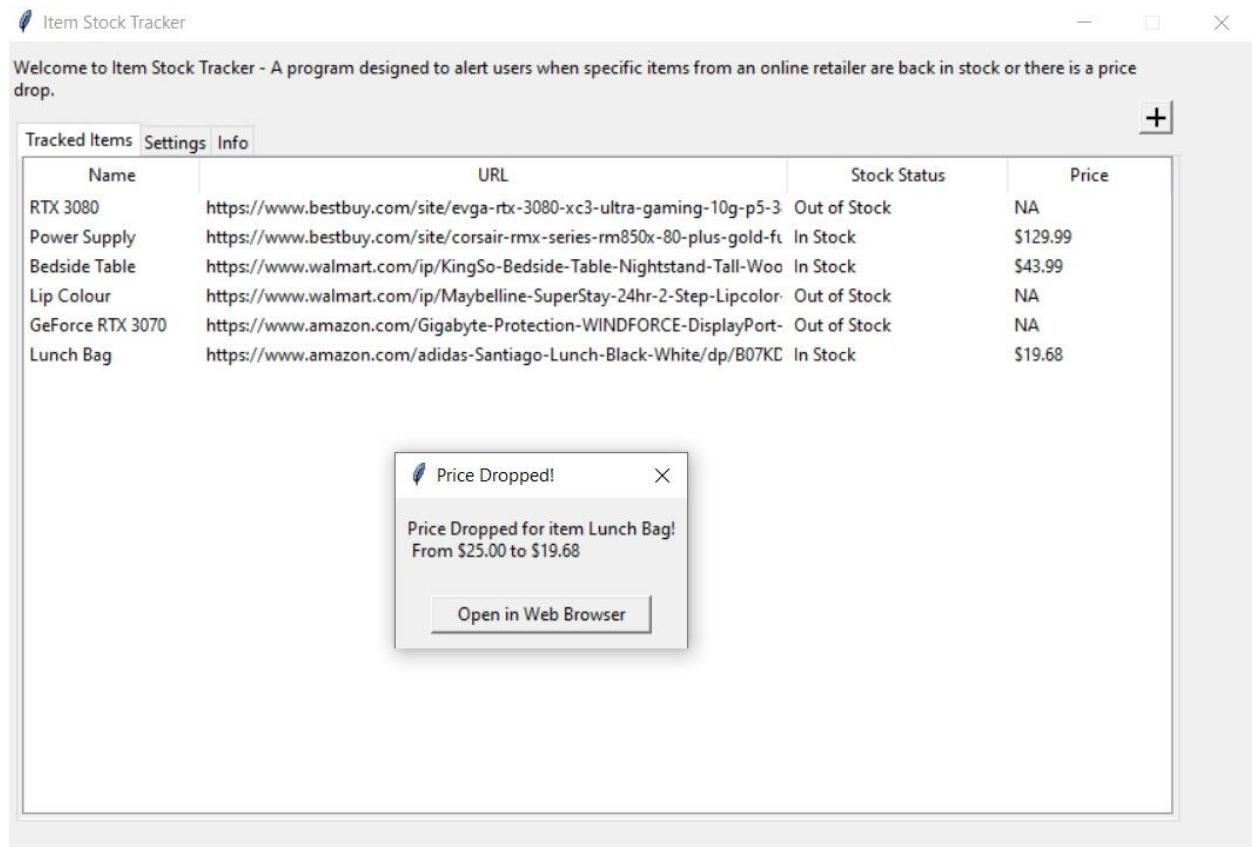


As part of our project 3 work, we have improved this project significantly from where it was left by the previous team. We have made multiple changes to the repo by changing the whole UI and adding new features. Below are some of the major enhancements achieved by our team.

## 1. Feature Enhancements

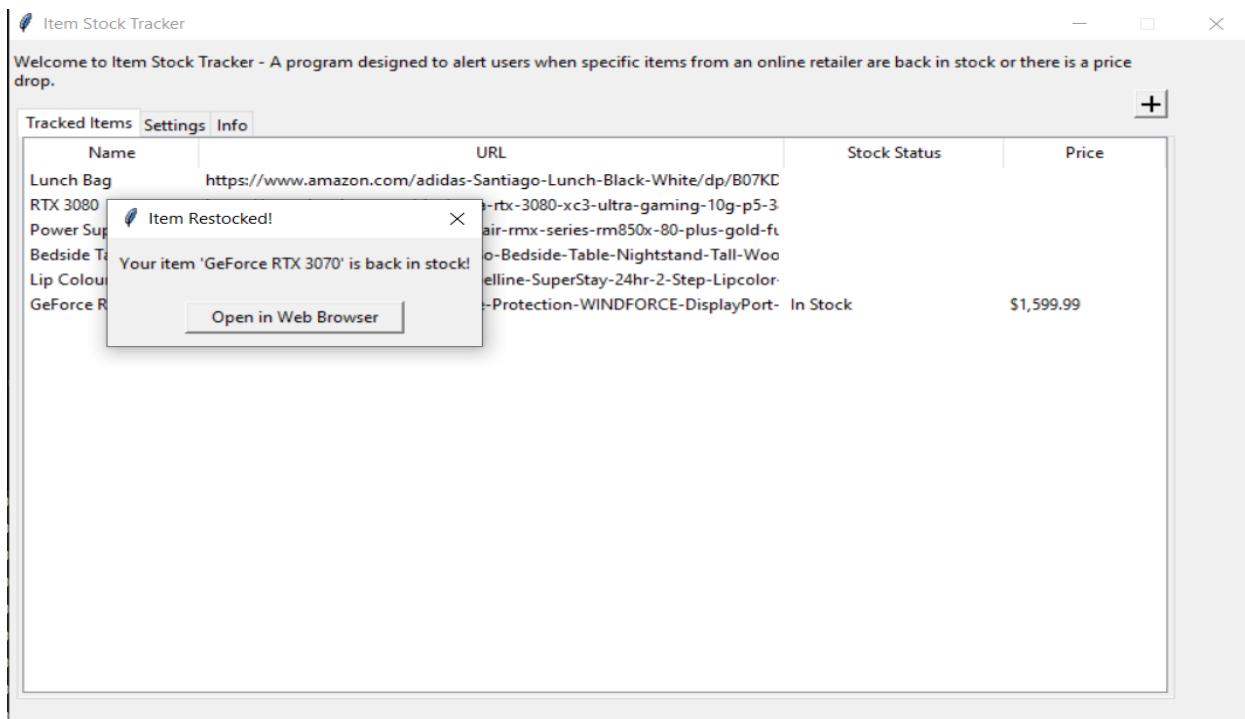
### Price Drop Alert

If a user wants to purchase certain products but wants to wait until the price is lowered, they can make use of our brand new “Price Drop Alert” feature. In this feature, We created triggers for the products that change their price from high to low. As shown in the example, A product named “Lunch Bag” changed its price from \$25.00 to \$19.68.

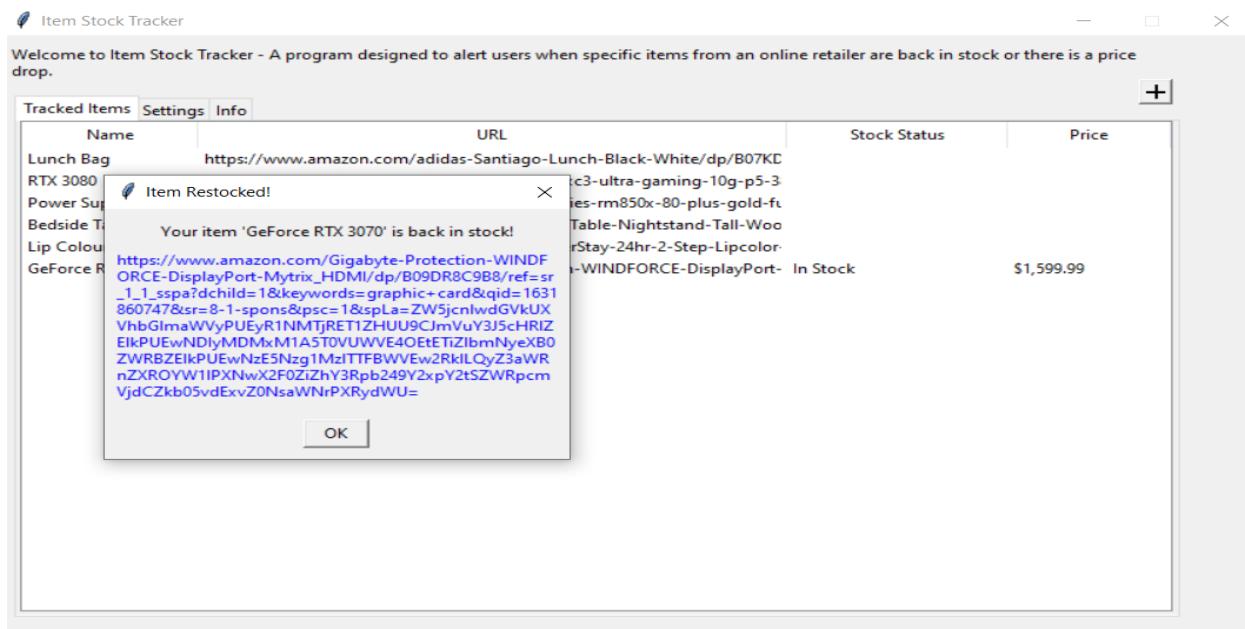


## Item Restock Alert

For out of stock items, users no longer have to check the webpages again and again. Our new feature will give users an item stock alert which will save their time. Whenever the item is back in stock the trigger will appear as shown below:



It also provides the checkout link of that item in the trigger popup.

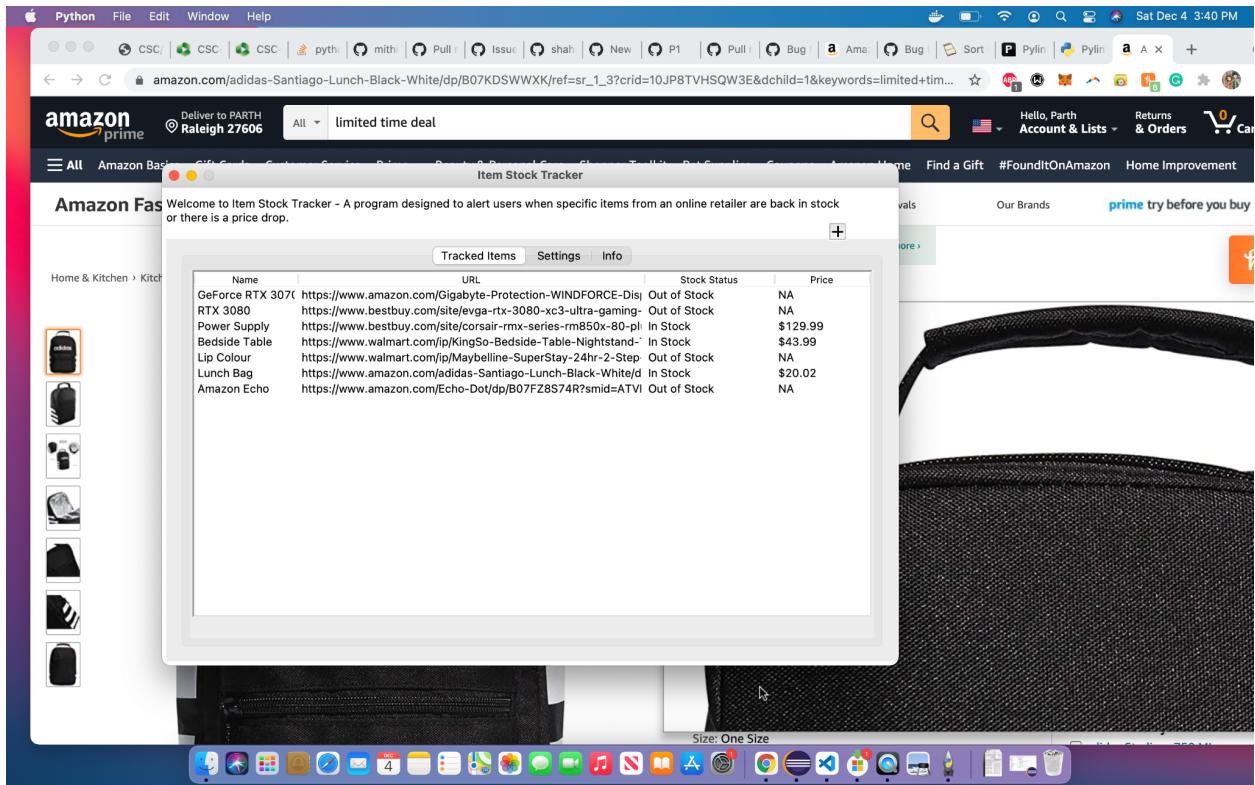


## Show Discounted Price

Earlier the scraper was fetching the real price of the product rather than the discounted price. Now if the e-commerce website adds new discounted prices, it will be shown on our dashboard. Earlier it was throwing a null value error.

The screenshot shows an Amazon product page for an Adidas Santiago Insulated Lunch Bag. The bag is black with the 'adidas' logo in white. On the left, there's a sidebar with small thumbnail images of various lunch bags. The main product image is large. To the right, product details are listed: \$20.02 (prime), FREE One-Day & FREE Returns, and FREE delivery tomorrow. It has 6,469 ratings and 29 answered questions. The item is labeled as #1 Best Seller in Sports Fan Lunch Boxes. Below the price, it says 'You Save: \$4.98 (19%)'. It notes that the item may be available at a lower price from other sellers, potentially without free Prime shipping. A color palette shows Black/White, and a size section indicates 'One Size'. At the bottom, there are 'Add to Cart' and 'Buy Now' buttons, along with a 'Secure transaction' note.

The screenshot shows a GitHub pull request titled 'Item Stock Tracker' with the commit message 'Here is the'. The pull request details a table of tracked items with columns for Name, URL, Stock Status, and Price. The items listed are Bedside Table, Lip Colour, Lunch Bag, Amazon Echo, GeForce RTX 3070, RTX 3080, and Power Supply. The status for most items is 'Out of Stock' or 'NA', except for the Power Supply which is 'In Stock' at \$129.99. To the right of the pull request, there's an 'Insights' panel with sections for Collaborator, Assignees (No one—assign yourself), Labels (None yet), Projects (None yet), Milestone (No milestone), Linked pull requests (Successfully merging a pull request may close this issue. None yet), and Notifications (Customize). A note at the bottom of the pull request says 'Remember, contributions to this repository should follow its contributing guidelines and code of conduct.'



## Amazon Scraper

We added a new scrapper for Amazon. Earlier we had scrapers for BestBuy and Walmart. The new Scrapper will help the users to track items from Amazon webpage.

<https://github.com/shahrk/ItemStockTracker/blob/main/code/AmazonScraper.py>

## Analytics page

We can show a price chart for a particular period in history. For this, the app would have to keep running on a server.

## **2. UI Enhancements**

We created a new Web based app for our project. The desktop version was tedious to install and maintain. It is much better for users to have a web-based version. We created a simple UI where users can log in and add items to track. They will be notified via email if/when the stock/price of the products they add changes. The plan for UI implementation looked like this:

Phase 1. Create a web UI for analytics page

Phase 2. Create a price tracker for the items , this will track the price items on that webpage.

Phase 3. If the prices or stock are changed, users can see it live on the tracking dashboard or will be notified via email.



## **3. Bug Fixes**

We have fixed several errors during the course of our projects. Some bugs were related to deployment and some were basic bugs in code. Below is the list of bugs that we fixed in this new version v3.0

<input type="button" value="Filters ▾"/>	<input type="text" value="is:issue is:closed label:bug"/>	<input type="button" value="Labels 9"/>	<input type="button" value="Milestones 0"/>	<input type="button" value="New issue"/>
<input checked="" type="checkbox"/> Clear current search query, filters, and sorts				
<input type="checkbox"/>	0 Open	✓ 5 Closed	Author ▾	Label ▾
<input type="checkbox"/>	<input checked="" type="checkbox"/> Bug fix 2: Fetch discounted price <span style="background-color: red; border-radius: 50%; padding: 2px 4px;">bug</span>			<input type="button" value="5"/>
	#13 by Parth59 was closed 1 hour ago			
<input type="checkbox"/>	<input checked="" type="checkbox"/> Bug Fix 1:- Issue when Opening a Restocked Item webpage. <span style="background-color: red; border-radius: 50%; padding: 2px 4px;">bug</span>			<input type="button" value="5"/>
	#12 by Parth59 was closed 2 hours ago			
<input type="checkbox"/>	<input checked="" type="checkbox"/> App not working when notification is triggered <span style="background-color: red; border-radius: 50%; padding: 2px 4px;">bug</span>			<input type="button" value="1"/>
	#6 by shahrk was closed 14 days ago			
<input type="checkbox"/>	<input checked="" type="checkbox"/> The scraper doesn't work for amazon.com products <span style="background-color: red; border-radius: 50%; padding: 2px 4px;">bug</span>			<input type="button" value="3"/>
	#2 by shahrk was closed 14 days ago			
<input type="checkbox"/>	<input checked="" type="checkbox"/> Unable to setup app on MacOS <span style="background-color: red; border-radius: 50%; padding: 2px 4px;">bug</span>			<input type="button" value="2"/>
	#1 by shahrk was closed 14 days ago			

## 1.App not working when notification is triggered

When we downloaded the project initially, we found that there was a major error in the functioning of the triggers. The GUI stops refreshing whenever the notification is triggered. The error message was shown in the console whenever notifications are pushed via trigger.

The fix is been provided in this commit:

<https://github.com/shahrk/ItemStockTracker/commit/85fdd753c560badb0c0e9dedaae8c54f5e3bf97d>

## 2. The support for MAC OS

The app was not deployable on mac os. There were several complications arising due to the Application initially developed using Windows.

```
rajshah@Rajs-MacBook-Air ~ % /Users/rajshah/Documents/se_2021/ItemStockTracker/build/dist/ItemStockTrackerMac/ItemStockTrackerMac ; exit;
Traceback (most recent call last):
  File "GUI.py", line 613, in <module>
    File "GUI.py", line 79, in __init__
      File "tkinter/__init__.py", line 4063, in __init__
        File "tkinter/__init__.py", line 4008, in __init__
    tkinter.TclError: couldn't open "data/plus.gif": no such file or directory
[4106] Failed to execute script 'GUI' due to unhandled exception: couldn't open "data/plus.gif": no such file or directory
[4106] Traceback:
(null)
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[Process completed]
```

These complications were creating a host of issues on MACOS. Thus we created a script to be executed on MAC to set up the environment and Maintain cross-compatibility. On top of it, we also added the automatic homebrew installation script in the fix.

The fix is been provided in this commit:

<https://github.com/shahrk/ItemStockTracker/commit/e237d4a94449fae9f2aaa80ce66527ab42490259>

### 3. Fixed SendEmail.py

Earlier, the code was using hardcoded email id/password to send price alerts. This now is fixed and now it sends the alerts using environmental IDs.

This was fixed in this commit:

<https://github.com/shahrk/ItemStockTracker/commit/29ffb8710287a0f8fba1780009d21dfe2bafec69>

### **4. Added new tools**

We added new code coverage, line counter, code style checker and code formatter tools. We used pylint for performing code style checks. We used Black Formatter for checking code format.

The screenshot shows a code editor interface with several tabs and panes.

**SOURCE CONTROL** pane:

- Shows a tree view of the repository structure under "ItemStockTracker".
- Shows changes for various files like setup.py, .gitignore, and trunk.yaml.

**.gitignore (Untracked) M** tab:

```
1 # Enter to commit on 'parthk'
```

**tracker.txt (Working Tree) M** tab:

```
1 version: 0.1
2 cli:
3 | version: 0.3.1-beta
4 lint:
5 enabled:
6 | - actionlint@1.6.4
7 | - black-py@21.10b0
8 | - flake8@3.9.2
9 | - gitleaks@7.6.1
10 | - isort@5.9.3
11 | - markdownlint@0.29.0
12 | - prettier@2.4.1
13 | - shellcheck@0.7.2
14 | - shfmt@3.4.0
15
```

**trunk.yaml (Index) M** tab:

```
1 version: 0.1
2 cli:
3 | version: 0.3.1-beta
4 lint:
5 enabled:
6 | - actionlint@1.6.4
7 | - black-py@21.10b0
8 | - flake8@3.9.2
9 | - gitleaks@7.6.1
10 | - isort@5.9.3
11 | - markdownlint@0.29.0
12 | - prettier@2.4.1
13 | - shellcheck@0.7.2
14 | - shfmt@3.4.0
15
```

**PROBLEMS**, **OUTPUT**, and **TERMINAL** panes:

**TERMINAL** pane:

```
SimStudents-MacBook-Pro:ItemStockTracker simstudent$ pwd/Users/simstudent/Desktop/SE/ItemStockTracker
SimStudents-MacBook-Pro:ItemStockTracker simstudent$ trunk check
Checking 100% [=====] 37/37 46.5s
```

**AUTOFIXES** pane:

```
install.sh
1:1 high Incorrect formatting shfmt
4 | pyenv global 3.9.8
5 | brew install pyinstaller
6 | pip3 install pyinstaller
| pip3 install pyinstaller

- Apply autofix (Y/n): Y
Autofix applied.

setup.py
1:1 high Incorrect formatting black-py
35 |     "setuptools",
36 |     "sphinx",
37 |     "sphinx-rtd-theme"
38 |     ],
39 |     "sphinx-rtd-theme",
| ],
39 | )

- Apply autofix (Y/n): Y
Autofix applied.

Re-checking autofixed files...
```

Bottom status bar:

```
x 8 parthk* Python 3.8.9 64-bit 0 0 △ 0 Ln 1, Col 1 Spaces: 2 UTF-8 YAML ⚡ Q
```