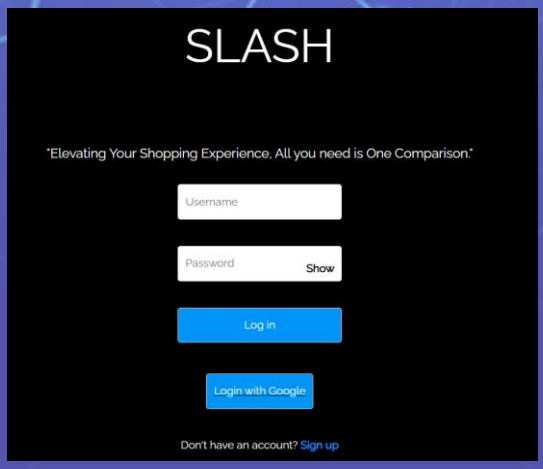
Scan to see a live demo!

SLASH

Do you love shopping? Are you in search of some good deals while shopping online?! **Slash** is here to help you look for the best deals!

Slash is a publicly accessible web API framework that allows one to scrape the most popular e-commerce websites to get the best deals on the searched items across multiple e-commerce websites. Currently supported websites include Amazon, Walmart, Target, BestBuy, Costco and EBay.



Flask
web development, one drop at a time

bcrypt

Features Added

- **OAuth Login Integration**: Enabled secure and streamlined user login through Google OAuth, enhancing user experience and security.
- **Password Hashing**: Implemented bcrypt password hashing to ensure secure storage of user credentials.
- **Export to CSV**: Added functionality to export search results to CSV for easy data access and analysis.
- **Multi-Currency Support**: Introduced multi-currency handling, allowing users to view product prices in their preferred currency.
- **UI Improvements**: Redesigned the user interface for a more intuitive and visually appealing experience.
- **Expanded Test Coverage**: Increased test coverage by writing additional test cases, ensuring reliable and robust functionality.
- **Enhanced Product Search Filters**: Added filters to allow users to refine search results by specific websites for targeted browsing.

Future Improvements

- **Add more Oauth login options like GitHub, Facebook, etc.**
- **Improve the Wishlist feature**
- **Database Integration: Add database for further enhancements to the project**
- **Price Chart Visualization: Introduce a visual representation of price trends for products.**
- **Adding support for more merchant websites like Temu**
- **Add feature to compare product specifications**
- **Enhance security by adding features such as two-factor authentication, etc.**

Group 82
Team Members: Atharva Pansare (aspansar), Ashwin Satpute (aasatput), Soham Patil (sspatil6)

Scan to view repo!

Why should YOU pick this? Scan to see!

CSC 510: Project 2

Team Number: 82

Team members:
Atharva Pansare (aspansar)

Ashwin Satpute (aasatput)

Soham Patil (sspatil6)

Link to repo: https://github.com/CSC510-SE-Fall2024/Team-82_Project-2

Sum of Self Assessment: 162

Note	Self-Assessment	Evidence
Workload is spread over the whole team (one team member is often Xtimes more productive than the other, but nevertheless, here is a track record that everyone is contributing a lot)	3	
Number of commits: by different people	3	GitHub repo link
Issues reports: there are many	3	GitHub repo link
Issues are being closed	3	
Docs: doco generated, format not ugly	3	GitHub repo docs
Docs: what: point descriptions of each class/function (in isolation)	3	GitHub repo docs
Docs: how: for common use cases X,Y,Z mini-tutorials showing worked examples on how to do X,Y,Z	3	GitHub repo docs
Docs: why: docs tell a story, motivate the whole thing, deliver a punchline that makes you want to rush out and use the thing	3	GitHub repo docs
Docs: short video, animated, hosted on your repo. That convinces people why they want to work on your code.	3	GitHub repo docs

Use of version control tools	3	
Test cases exist	3	GitHub repo tests
Test cases are routinely executed	3	
Issues are discussed before they are closed	3	
Chat channel: exists	3	https://discord.com/invite/UF5Hr2dW
Test cases: a large proportion of the issues related to handling failing cases.	2	
Evidence that the whole team is using the same tools: everyone can get to all tools and files	3	GitHub repo link
Evidence that the whole team is using the same tools (e.g. config files in the repo, updated by lots of different people)	3	
Evidence that the whole team is using the same tools (e.g. tutor can ask anyone to share screen, they demonstrate the system running on their computer)	3	
Evidence that the members of the team are working across multiple places in the code base	3	
Short release cycles	3	GitHub repo link
The file .gitignore lists what files should not be saved to the repo.	3	GitHub repo link
The file INSTALL.md lists how to install the code	3	GitHub repo link
The file LICENSE.md lists rules of usage for this repo	3	GitHub repo link
The file CODE-OF-CONDUCT.md lists rules of behavior for this repo; e.g. see example	3	GitHub repo docs
The file CONTRIBUTING.md lists coding standards and lots of tips on how to extend the system without screwing things up; e.g. see example	3	GitHub repo docs

The file README.md contains all the following	3	GitHub repo link
Video	3	
DOI badge: exists. To get a Digitial Object Indentifier, regiser the project at Zenodo. DOI badges look like this: Zenodo doi badge	3	
Badges showing your style checkers	3	
Badges showing your code formatters.	3	
Badges showing your syntax checkers.	3	
Badges showing your code coverage tools	3	
Badges showing any other Other automated analysis tools	3	
Does your website and documentation provide a clear, high-level overview of your software?	3	
Does your website and documentation clearly describe the type of user who should use your software?	3	
Do you publish case studies to show how your software has been used by yourself and others?	3	
Is the name of your project/software unique?	3	
Is your project/software name free from trademark violations?	2	
Is your software available as a package that can be deployed without building it?	3	
Is your software available for free?	3	
Is your source code publicly available to download, either as a downloadable bundle or via access to a source code repository?	3	

Is your software hosted in an established, third-party repository like GitHub?	3	
Is your documentation clearly available on your website or within your software?	3	
Does your documentation include a "quick start" guide, that provides a short overview of how to use your software with some basic examples of use?	2	
If you provide more extensive documentation, does this provide clear, step-by-step instructions on how to deploy and use your software?	3	
Do you provide a comprehensive guide to all your software's commands, functions and options?	3	
Do you provide troubleshooting information that describes the symptoms and step-by-step solutions for problems and error messages?	3	
If your software can be used as a library, package or service by other software, do you provide comprehensive API documentation?	3	
Do you store your documentation under revision control with your source code?	2	
Do you publish your release history e.g. release data, version numbers, key features of each release etc. on your web site or in your documentation?	3	
Does your software describe how a user can get help with using your software?	3	
Does your website and documentation describe what support, if any, you provide to users and developers?	3	
Does your project have an e-mail address or forum that is solely for supporting users?	3	

Are e-mails to your support e-mail address received by more than one person?	3	
Does your project have a ticketing system to manage bug reports and feature requests?	2	
Is your project's ticketing system publicly visible to your users, so they can view bug reports and feature requests?	2	