

Github link: <https://github.com/SE-Fall-2024-Team-69/slash>

Ayush Gupta- agupta86@ncsu.edu

Yaswanth Baddi - ybaddi@ncsu.edu

Yaswanth Mullamuri - vmullam@ncsu.edu

Poster:



VIDEO

Group 69

Yaswanth Baddi
Yaswanth Mullamuri
Ayush Gupta

What is slash?

Slash is an innovative open-source application that transforms the online shopping experience by aggregating product information from major retail websites such as Walmart, Target, Costco, and eBay into a centralized platform. By consolidating data, Slash enables users to compare prices, read reviews, and make informed purchasing decisions, streamlining their shopping journey.



Elevated User Experience

We've transformed the user experience by introducing a sleek, modern interface designed to be both visually appealing and highly intuitive. The updated layout enhances usability, making it effortless for users to explore features, discover content, and navigate seamlessly across the website. This redesign prioritizes simplicity, ensuring users can find what they need quickly while enjoying a more engaging and responsive interface throughout their entire journey.

Interactive Comments Section for Community Engagement

Alongside security enhancements, this update brings an exciting new feature to our product pages: an interactive comments section. This addition empowers users to engage directly by asking questions, sharing feedback, and offering helpful insights to others interested in the product. Whether users want to discuss their experiences or seek advice, this section fosters a sense of community and connection, making each product page a dynamic space for valuable interactions and shared knowledge.

Enhanced Security and Google Account Login

In this latest upgrade, we've prioritized user security and convenience by implementing robust password encryption techniques. Our new system safeguards user data with advanced encryption, ensuring that login credentials remain protected. Additionally, we've integrated seamless Google account login, making it faster and easier to access the platform with the security and familiarity of Google's verification process. Together, these improvements provide a secure, streamlined, and modernized login experience compared to previous methods.

Working Screenshot



Future Enhancements:

Slash can incorporate coupon scraping to help users save even more. We plan to add more scrapers to cover a wider range of e-commerce sites and enable specification comparisons within the wishlist to simplify product selection. Additionally, including shipping charges in price comparisons would allow users to see the total cost, making Slash an even more complete tool for finding the best deals.

Self - assessment:

Total number: 193

Notes	self-assessment	evidence							
Workload is spread over the whole team (one team member is often Xtimes more productive than the others... but nevertheless, here is a track record that everyone is contributing a lot)		2	https://github.com/SE-Fall-2024-Team-69/slash/pulse						
Number of commits		2	https://github.com/SE-Fall-2024-Team-69/slash/pulse						
Number of commits: by different people		2	https://github.com/SE-Fall-2024-Team-69/slash/commits/main/						
Issues reports: there are many		2	https://github.com/SE-Fall-2024-Team-69/slash/commits/main/						
Issues are being closed		2	https://github.com/search?q=repo%3ASE-Fall-2024-Team-69%2Fslash+issues&type=issues						
Docs: doco generated, format not ugly		1	https://github.com/search?q=repo%3ASE-Fall-2024-Team-69%2Fslash+issues&type=issues						
Docs: what: point descriptions of each class/function (in isolation)		2	https://github.com/SE-Fall-2024-Team-69/slash						
Docs: how: for common use cases X,Y,Z mini-tutorials showing worked examples on how to do X,Y,Z		3	https://github.com/SE-Fall-2024-Team-69/slash						
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	doc page ehttps://github.com/SE-Fall-2024-Team-69/slash/blob/main/README.mdtries						
Docs: short video, animated, hosted on your repo. That convinces people why they want to work on your code.		2	https://github.com/SE-Fall-2024-Team-69/slash/blob/main/README.md						
Test cases exist		2	https://github.com/SE-Fall-2024-Team-69/slash/blob/main/README.md						
Test cases are routinely executed		3	https://github.com/SE-Fall-2024-Team-69/slash/tree/main/tests						
Evidence that the whole team is using the same tools: everyone can get to all tools and files		1	https://github.com/SE-Fall-2024-Team-69/slash/actions						
Evidence that the whole team is using the same tools (e.g. config files in the repo, updated by lots of different people)		2	https://github.com/SE-Fall-2024-Team-69/slash/blob/main/requirements.txt						
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	https://github.com/SE-Fall-2024-Team-69/slash/blob/main/requirements.txt						
Evidence that the members of the team are working across multiple places in the code base		2	https://github.com/SE-Fall-2024-Team-69/slash/blob/main/requirements.txt						
Short release cycles		3	https://github.com/SE-Fall-2024-Team-69/slash/commits/main/						
The file INSTALL.md lists how to install the code		2	https://github.com/SE-Fall-2024-Team-69/slash/commits/main/						
The file LICENSE.md lists rules of usage for this repo		1	https://github.com/SE-Fall-2024-Team-69/slash/blob/main/INSTALL.md						
The file CODE-OF-CONDUCT.md lists rules of behavior for this repo; e.g. see example		2	https://github.com/SE-Fall-2024-Team-69/slash/blob/main/LICENSE						
The file README.md contains all the following		3	https://github.com/SE-Fall-2024-Team-69/slash?tab=coc-ov-file						
Video		2	https://github.com/SE-Fall-2024-Team-69/slash/blob/main/README.md						
DOI badge: exists. To get a Digital Object Identifier, register the project at Zenodo. DOI badges look like this:		3	https://drive.google.com/file/d/1Efehn_X6z4WYxkIG2uGirg7ytlan8HI/view?usp=drive_link						
Badges showing your style checkers		0	https://github.com/SE-Fall-2024-Team-69/slash/blob/main/README.md						
Badges showing your code formatters.		1	https://github.com/SE-Fall-2024-Team-69/slash/blob/main/README.md						
Badges showing your syntax checkers.		2	https://github.com/SE-Fall-2024-Team-69/slash/blob/main/README.md						
Badges showing your code coverage tools		3	https://github.com/SE-Fall-2024-Team-69/slash/blob/main/README.md						
Does your website and documentation provide a clear, high-level overview of your software?		3	https://github.com/SE-Fall-2024-Team-69/slash/blob/main/README.md						
Does your website and documentation clearly describe the type of user who should use your software?		2							
Do you publish case studies to show how your software has been used by yourself and others?		0							
Is the name of your project/software unique?		3							
Is your project/software name free from trademark violations?		3							
Is your software available as a package that can be deployed without building it?		2							
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?		2							
Is your software hosted in an established, third-party repository like GitHub (https://github.com), BitBucket (https://bitbucket.org), LaunchPad (https://launchpad.net) or SourceForge (https://sourceforge.net)?		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?		3							
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?		2							
Do you provide troubleshooting information that describes the symptoms and step-by-step solutions for problems and error messages?		0							
If your software can be used as a library, package or service by other software, do you provide comprehensive API documentation?		0							
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?		1							
Does your software describe how a user can get help with using your software?		2							
Does your website and documentation describe what support, if any, you provide to users and developers?		1							
Does your project have an e-mail address or forum that is solely for supporting users?		2							
Are e-mails to your support e-mail address received by more than one person?		1							
Does your project have a ticketing system to manage bug reports and feature requests?		1							
Is your project's ticketing system publicly visible to your users, so they can view bug reports and feature requests?		1							
Is your software's architecture and design modular?		2							
Does your software use an accepted coding standard or convention?		3							
Does your software allow data to be imported and exported using open data formats?e.g. GIF, SVG, HTML, XML, tar, zip, CSV, JSON, NetCDF, or domain specific ones		3							
Does your software allow communications using open communications protocols? e.g. HTTP, FTP, XMPP, SOAP over HTTP, or domain-specific ones		3							

Is your software cross-platform compatible? e.g. does it run under two or more of Windows, Unix/Linux and Mac OS X, or can be used from within two or more of Internet Explorer, Chrome, Firefox and Safari?	3								
Does your software adhere to appropriate accessibility conventions or standards?	3								
Does your documentation adhere to appropriate accessibility conventions or standards?	3								
Is your source code stored in a repository under revision control?	3								
Is each source code release a snapshot of the repository?	1								
Are releases tagged in the repository?	1								
Is there a branch of the repository that is always stable? (i.e. tests always pass, code always builds successfully)	3								
Do you back-up your repository?	3								
Do you provide publicly-available instructions for building your software from the source code?	1								
Can you build, or package, your software using an automated tool? e.g. Make (https://www.gnu.org/software/make/), ANT (http://ant.apache.org/), Maven (https://maven.apache.org/), CMake (https://cmake.org/), Python setuptools (https://pypi.python.org/pypi/setuptools), or R package tools (https://cran.r-project.org/doc/manuals/r-devel/R-exts.html)	3								
Do you provide publicly-available instructions for deploying your software?	3								
Does your documentation list all third-party dependencies?	3								
Does your documentation list the version number for all third-party dependencies?	3								
Does your software list the web address, and licences for all third-party dependencies and say whether the dependencies are mandatory or optional?	3								
Can you download dependencies using a dependency management tool or package manager?	1								
Do you have tests that can be run after your software has been built or deployed to show whether the build or deployment has been successful?	1								
Do you have an automated test suite for your software?	2								
Do you have a framework to periodically (e.g. nightly) run your tests on the latest version of the source code?	1								
Do you use continuous integration, automatically running tests whenever changes are made to your source code?	1								
Are your test results publicly visible?	3								
Are all manually-run tests documented	1								
Does your project have resources (e.g. blog, Twitter, RSS feed, Facebook page, wiki, mailing list) that are regularly updated with information about your software? e.g. release announcements, publications, workshops, conference presentations	1								
Does your website state how many projects and users are associated with your project?	2								
Do you provide success stories on your website?	1								
Do you list your important partners and collaborators on your website?	1								
Do you list your project's publications on your website or link to a resource where these are available?	0								
Do you list third-party publications that refer to your software on your website or link to a resource where these are available?	0								
Can users subscribe to notifications to changes to your source code repository?	0								
If your software is developed as an open source project (and, not just a project developing open source software), do you have a governance model?	3								
Do you accept contributions (e.g. bug fixes, enhancements, documentation updates, tutorials) from people who are not part of your project?	3								
Do you have a contributions policy?	2								
Is your contributions' policy publicly available?	2								
Do contributors keep the copyright/IP of their contributions?	1								
Does your website and documentation clearly state the copyright owners of your software and documentation?	2								
Does each of your source code files include a copyright statement?	1								
Does your website and documentation clearly state the licence of your software?	1								
Is your software released under an open source licence?	3								
Is your software released under an OSI-approved open-source licence?	2								
Does each of your source code files include a licence header?	3								
Do you have a recommended citation for your software?	2								
Does your website or documentation include a project roadmap (a list of project and development milestones for the next 3, 6 and 12 months)?	2								
Does your website or documentation describe how your project is funded, and the period over which funding is guaranteed?	1								
Do you make timely announcements of the deprecation of components, APIs, etc.?	2								