Code Vault Final Report

NEIL AKALWADI, Virginia Tech, USA
ANIRUDH CHINTA, Virginia Tech, USA
RIYA NAIR, Virginia Tech, USA
KUHU PANDIT, Virginia Tech, USA
ERIC RIPPEY, Virginia Tech, USA

Developers frequently face the inefficiency of writing redundant code for common tasks, especially when solutions already exist but are not easily accessible. Our enhanced Code Vault platform now incorporates advanced filtering options using tags, improved search functionality, and intuitive user interfaces, based on feedback from our initial design. The web-based repository enables developers to efficiently share, discover, and reuse optimized code snippets in various programming languages. By emphasizing seamless integration, team collaboration, and scalability, Code Vault not only simplifies development workflows but also supports a culture of shared learning and innovation. These improvements significantly enhance the productivity and accessibility of reusable code for developers of all levels.

ACM Reference Format:
Neil Akalwadi, Anirudh Chinta, Riya Nair, Kuhu Pandit, and Eric Rippey. 2024. Code Vault Final Report. 1, 1 (December 2024), 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

- Problem Definition
  In the software development life cycle, repetitive tasks often lead to inefficiencies. Developers spend substantial time rewriting solutions for problems that have already been solved, resulting in delayed project timelines and increased costs. While this is especially true when programming in lower level languages such as C, C++, and FORTRAN, it occurs in all languages. Despite the existence of platforms like GitHub, these tools often lack tailored solutions for code reuse in small, specific scenarios.

- Motivating example
  Consider a software engineer tasked with implementing a commonly used feature, such as user authentication, across multiple projects. Without an efficient way to retrieve and adapt previously written code, they would need to write the functionality from scratch or manually sift through older projects. Code Vault eliminates this bottleneck by providing a centralized platform where developers can quickly find and integrate tested code snippets, saving hours of work and reducing errors. Apply the re-usability concept across reusable snippets such as data structures, file parsing, network controllers, and more, and the time

Authors' addresses: Neil Akalwadi, aneil@vt.edu, Virginia Tech, Blacksburg, Virginia, USA; Anirudh Chinta, anirudhchinta03@vt.edu, Virginia Tech, Blacksburg, Virginia, USA; Riya Nair, nriya@vt.edu, Virginia Tech, Blacksburg, Virginia, USA; Kuhu Pandit, kuhupandit@vt.edu, Virginia Tech, Blacksburg, Virginia, USA; Eric Rippey, erippey3@vt.edu, Virginia Tech, Blacksburg, Virginia, USA.

ACM XXXX-XXXX/2024/12-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

to get projects off the ground diminishes rapidly.

- Proposed Solution
  Code Vault is a comprehensive, user-friendly web-based repository where developers can store, share, and retrieve reusable code snippets. The platform supports multi-language capabilities, advanced tagging, and search features, catering to the needs of both individual developers and collaborative teams.

1.1 background

In this section, define any key terms and key concepts that will be used throughout your work (if applicable) To fully understand the value and purpose of Code Vault, it is essential to define some key terms and concepts related to the project:
Code Snippet: A code snippet is a small, reusable piece of code that solves a specific problem or implements a commonly used functionality. Examples include code for user authentication, sorting algorithms, or API calls. Code snippets are particularly valuable because they save developers time, reduce errors, and improve consistency across projects.
Tagging System: A tagging system allows developers to categorize snippets using descriptive keywords, such as "authentication," "database query," or "sorting algorithm." By tagging code snippets appropriately, users can efficiently filter and search through the repository to locate relevant solutions without manual scanning.
Search Functionality: Effective search functionality enables users to retrieve relevant code snippets using keywords, tags, or other filters. Unlike searching through entire code repositories, the search in Code Vault is optimized for locating small, modular snippets quickly. This approach improves productivity and reduces the time spent searching for reusable solutions.
Code Repository: A code repository is a centralized storage system where developers can manage, organize, and share code. Platforms like GitHub are commonly used to host repositories containing entire projects or applications. However, searching for small, specific solutions within a large repository can be inefficient. Code Vault focuses on hosting only modular code snippets, making it easier to retrieve targeted solutions.
Multi-Language Support: Modern development teams often use multiple programming languages depending on project needs. For example, front-end developers may use JavaScript, while back-end developers may use Python or Java. Code Vault supports multi-language snippets, ensuring that developers working across different tech stacks can benefit equally.

2 RELATED WORK

Several tools have emerged to address coding efficiency and minimize repetitive tasks. Notable competitors include: GitHub Gist- A web-based service for sharing and managing code snippets, notes, and small files, with version control. Stack Overflow Code Snippets is another competitor that enables users to share and discuss code examples within programming Q&A contexts. Code Vault distinguishes itself with additional search features such as tagging. The end goal for this is a platform is re-usability and simplicity.

3 IMPLEMENTATION

3.1 backend

Our website serves content using a Flask backend. This approach was used for its balance of features, simplicity, and familiarity to our team. Flask allowed our team to stay organized as well, with built in modularity in the form of blueprints. Each piece of funtionality had its own blueprint, from authentication, to file handling, to searching/browsing. Alongside flask, we have a few other things running. These include a Postgress database, used for tagging, a ElasticSearch engine, used for searching by keywords, and a Redis database, used for session information.

3.2 frontend

As of this moment, the platform serves HTML, with static CSS and embedded java script. While it is not the most elegant approach, it worked for a MVP. A good future step would be to improve the front end by implementing a react layer on top of the Flask backend. This would also lessen the number of calls to the backend, putting more of the work on the client.

3.3 deployment

As of this current moment, our application is deployed using docker compose. A total of 4 containers, including the Flask instance, and the databases, are deployed together. Docker was chosen because it allows segmentation of the application from the rest of the computer. As of now, the application is being deployed on a personal server owned by one of our group members. A future plan would be to move the application to a more robust location with high availability.

While the site has only been up for a few months, our group has had time to learn about the maintenance of the site. First, it is important to periodically monitor the traffic that goes in and out of the site for suspicious activity. Checking logs on the network, and in the container itself are a good way to do that. Additionally, checking the health and status of all databases regularly is an important part of maintenance. Further, whenever our team encounters a bug, we report on it and make sure a fix gets pushed to production as soon as possible.

4 FUTURE PLANS

We have a four phase plan for future improvements that align with what we see as our biggest limitations at this moment. First, our code snippets are only located on the web, which is not where programmers develop most. Our future plans in phase one cover how users can discover, upload, and download our snippets. We want to make an executable to interface with the command line as well as a mobile application so users can discover on the go. Next, we feel our site still lacks many important security features. In phase two, we will implement security features such as encryption on all files stored from used, including user data, and code snippets. We also want continue to monitor our site for vulnerabilities and improve the fortitude of our website against malicious attacks. Third our site still lacks permissions. In phase three, we want to allow our users to make public and private repository options for more control over the code users post on our site. Lastly, our website does not scale well with high traffic. Phase four involves moving all of our services over to a high availability cluster running on something like Kubernetes.

the references are worth 5 points. to change them, edit the bibliography.bib page.

REFERENCES

[1] Discover gists. *Gist.* Accessed: 2024-09-27. n.d. URL: https://gist.github.com/starred.

[2] David Haney. *Introducing Runnable JavaScript, CSS, and HTML Code Snippets.* Accessed: 2024-09-27. Sept. 2014. URL: https://stackoverflow.blog/2014/09/16/introducing-runnable-javascript-css-and-html-code-snippets/.

Received 17 December 2024