Sahub - Stackoverflow and Comments integrations

André Oliveira*, Paulo Matos†, Pedro Filipe Oliveira†
*Instituto Politécnico de Bragança, Campus de Santa Apolónia, 5300-253 Bragança, Portugal
a64380@alunos.ipb.pt
†*Research Centre in Digitalization and Intelligent Robotics (CeDRI),*
Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC),
Instituto Politécnico de Bragança, Campus de Santa Apolónia, 5300-253 Bragança, Portugal
{pmatos, poliveira}@ipb.pt

2024 International Conference on Engineering and Emerging Technologies (ICEET) | 979-8-3315-3289-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICEET65156.2024.10913764

***Abstract*—This paper proposes a novel approach for connecting StackOverflow discussions and open source codebases through a graph database model in Neo4j. By encouraging developers to adopt a structured annotation framework using widely known tags such as @author, @description, @link, @tags, and optional fields such as @operatingSystem and @fileExtension, the paper will discuss how to link code snippets to specific technical questions. These annotations are then analyzed, creating an interconnected knowledge graph that can associate StackOverflow questions with real-world code examples and additional documentation pulled directly from code snippets.**

***Index Terms*—stackoverflow, JSDoc, Github, Git, comments, IPB, graph databases, Neo4j, software engineering**

I. Introduction

In the ever-evolving field of software development, access to practical, well-documented, and contextually relevant examples of code has become a cornerstone of efficient learning and problem-solving [1]. StackOverflow, as one of the largest online forums for programming questions [2], has played a pivotal role in democratizing access to this knowledge. However, the rapid growth in the volume of questions and answers has created new challenges, especially when it comes to finding contextually relevant code that fits specific issues faced by developers in real-world scenarios. While StackOverflow provides a wealth of information [3], it often lacks direct connections to open-source code repositories, where real-world examples could provide further context and insight [4].

Programming is a non.ending problem.solving adventure [1], historically, developers have relied on a combination of documentation, community discussions, and source code repositories like GitHub to learn from existing solutions and build upon previous work [2]. However, bridging the gap between a question raised on StackOverflow and a practical implementation found in an open-source codebase remains a persistent challenge. This gap is not only a barrier to efficient knowledge transfer but also a missed opportunity to deepen understanding by examining code within its actual context, which some studies have shown to be the best form of knowledge transfer [5].

979-8-3503-1692-6/23/$31.00 ©2024 IEEE

Another recurring issue is that, given StackOverflow's long history, there is a vast repository of questions and answers, many of which date back several years. As a result, some of the solutions provided may be unanswered, outdated, or obsolete due to evolving technology and programming standards [6]. Furthermore, the solutions found on StackOverflow may not always directly apply to specific contexts or unique challenges faced by users. In many cases, it can be challenging to adapt these generalized solutions for practical use in real-world applications, making them difficult to implement effectively.

To address this challenge, this paper proposes a new framework that creates a relationship knowledge graph that links StackOverflow discussions directly to real-world code examples hosted on open source platforms. Through this approach, we aim to foster a knowledge ecosystem where community members can more easily access code that is contextual and directly applicable to their questions. By combining StackOverflow's comprehensive repository of questions with real-world context found in open source projects, this framework has the potential to significantly improve the accessibility and usability of programming knowledge. And in the end will be created a google extension to test this approach and a simple website to submit a code with snippets.

The structure of this paper is as follows. The next section, "State of the Art," reviews similar projects that integrate StackOverflow and other community-driven platforms with development environments or codebases, highlighting their approaches and limitations. This is followed by the "Materials and Methods" section, which describes the methodology used to build the knowledge graph, covering data extraction from StackOverflow, annotation syntax inspired by JSDoc, and the process of matching keywords to link questions with code snippets. The "Results" section presents the outcomes of the project, illustrating the graph model with examples of connected posts and snippets. Finally, the paper concludes with a summary of contributions and suggestions for future enhancements, including the integration of machine learning models and potential extensions, such as a browser plugin or IDE integration to streamline code access directly from StackOverflow.

II. State of the art

A. Similar projects review

The integration of community-based question-and-answer platforms with collaborative coding environments has received considerable attention from the research community, as it holds potential to streamline developer knowledge-sharing and problem-solving. Platforms like StackOverflow and GitHub each play essential roles within the developer ecosystem: StackOverflow provides a forum for quickly accessible advice and solutions to technical challenges, while GitHub serves as a space for collaborative coding and open-source contributions. Yet, connecting these two resources to facilitate more contextually relevant code-based learning remains a challenge.

Below three projects are described.

1) StackOverflow and GitHub: Associations Between Software Development and Crowdsourced Knowledge: One significant study in this area, conducted by Vasilescu, Filkov, and Serebrenik [2], explores the associations between developer activities on StackOverflow and GitHub. Their research examines how participation in StackOverflow (particularly in asking and answering questions) correlates with productivity in GitHub, measured by the number of code commits. Their findings indicate that more active GitHub contributors tend to ask fewer questions on StackOverflow, while those who frequently answer questions often have higher commit rates. This relationship suggests that frequent answerers might also be more experienced developers who require less external assistance but are motivated to share their expertise with the community [2].

Moreover, the study observes a coordination between GitHub commit rates and StackOverflow participation, particularly for developers who actively engage in both communities. For these developers, bursts of activity on StackOverflow (such as answering questions) often coincide with increased GitHub commit activity. This dual engagement highlights a pattern where developers not only solve immediate technical challenges on StackOverflow but also apply this knowledge directly in open-source projects on GitHub. This study provides foundational insights into the dynamic interactions between collaborative coding and crowdsourced knowledge-sharing, illustrating the impact of community engagement on developer productivity across platforms [2].

2) Seahawk: Stack Overflow in the IDE: The Seahawk project by Ponzanelli, Bacchelli, and Lanza is a notable example of integrating StackOverflow into the IDE to support developers directly in their coding environment. Designed as an Eclipse plugin, Seahawk automates the process of querying StackOverflow based on the developer's active context within the IDE, reducing the need for manual searches. By automatically extracting relevant keywords from code entities, Seahawk formulates search queries, which return a ranked list of related discussions from StackOverflow. This integration helps developers by allowing them to view answers, import code snippets through drag-and-drop, and link StackOverflow discussions to code artifacts persistently within the IDE [7].

Seahawk addresses several challenges associated with accessing Q&A resources outside of the IDE. Traditionally, developers have to switch between their IDE and a web browser to search for answers, which disrupts focus and can hinder productivity. By keeping the information accessible within the development environment, Seahawk reduces context-switching and helps developers maintain their workflow. This project illustrates the productivity benefits of embedding community knowledge within the IDE, allowing developers to leverage "crowd knowledge" more effectively and enabling more collaborative and informed coding practices [7].

3) Mining StackOverflow to turn the IDE into a self-confident programming prompter: The Prompter plugin, designed for the Eclipse IDE, is another example of an integration aimed at enhancing developer productivity by embedding StackOverflow discussions within the coding environment. Developed by Ponzanelli et al., Prompter monitors a developer's actions and uses the current code context to automatically search for relevant discussions on StackOverflow. Once it finds discussions that meet a confidence threshold, it notifies the developer directly within the IDE, allowing for on-demand and contextually relevant guidance. The plugin ranks StackOverflow discussions based on factors like code similarity, textual relevance, and community metrics (e.g., user reputation), ensuring the recommendations align with the developer's current focus [8].

In evaluations, Prompter demonstrated that it could significantly improve task completion by providing relevant, context-sensitive recommendations that developers might otherwise spend time searching for manually. This approach minimizes workflow disruption and provides a streamlined way to access crowd-sourced knowledge without leaving the development environment. The plugin highlights how automated knowledge retrieval in IDEs can help developers solve problems efficiently, supporting both maintenance and new development tasks with relevant insights from the StackOverflow community [8].

III. Materials and Methods

This section describes the methodology used to connect StackOverflow posts with real-world code snippets through a structured graph-based model. The following subsections will provide details on StackOverflow data extraction, annotation syntax, relationship mapping, and future expansions.

A. StackOverflow data extraction

The data extraction process is a fundamental step in creating and validating the functionality of the relationship system proposed here. This project will use the StackOverflow API, which is specifically designed for developers to access the vast amount of data available on the platform. This API provides endpoints that allow questions, answers, and tags related to various programming topics to be retrieved, thus allowing relevant discussions to be selected for linking to code snippets. By leveraging this API, it will be possible to streamline data collection and keep the model up to date with the latest

questions and discussions on StackOverflow, ensuring that our graph remains relevant and reflects the latest trends in programming.

The StackOverflow API is flexible, offering several filters and query parameters to target specific content [3]. For example, it allows us to sort by activity, votes, and creation date, which helps to prioritize the most relevant or frequently accessed questions. In addition, it is possible to filter the results by specific tags, such as javascript or typescript, restricting the data to topics relevant to the available code snippets [9].

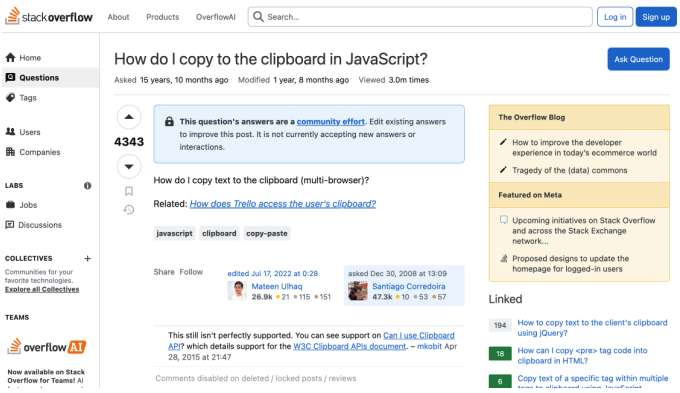

Fig. 1. Post example

For initial validation of its relevance, 1 StackOverflow post will be selected regarding a question about how to copy content (text) to the device's clipboard, using JavaScript, this post can be seen in the figure 1.

B. Snippet and sintaxe

JSDoc was chosen as the inspiration for Sahub's annotation format due to its wide adoption and familiarity within the developer community. Common tags such as @description, @tags and optional fields such as @link are used to provide metadata about the functionality and context of the code. All fields can be viewed in the table I. The @author, @description and @tags tags are mandatory, and the @fileExtension and @operatingSystem tags can be retrieved automatically based on the file and the user's computer.

TABLE I
SAHUB SYNTAX

Tag	Description	Required	Automatic
@author	Code author's name.	X	
@description	Brief code summary.	X	
@deprecated	Marks outdated code.		
@fileExtension	Expected file type.		X
@operatingSystem	Compatible OS(es).		X
@link	Link to usage guide.		
@tags	Relevant keywords.	X	

In the "StackOverflow data extraction" subsection, the post that will be used as an example was defined. A snippet of a code example that can be used to copy text to the user's clipboard will be used, using JavaScript and with the following parameters:

1) **Author**: André Saraiva
2) **Description**: Copy text to clipboard
3) **FileExtensions**: ts
4) **OperatingSystem**: MacOs
5) **Link**: https://github.com/example.js
6) **Tags**: copy, clipboard, text, typescript and javascript

Below you can see an example of this snippet in the figure 2

```
/**
 * @author André Saraiva
 * @description Copy text to clipboard
 * @fileExtension ts
 * @operatingSystem MacOs
 * @link https://github.com/andre2654/sahub/blob/main/snippets/clipboard.ts
 * @tags copy, clipboard, text, typescript, javascript
 */
export function copyToClipboard(text: string): void {
  const input = document.createElement('input')
  input.value = text
  document.body.appendChild(input)
  input.select()
  document.execCommand('copy')
  document.body.removeChild(input)
}
```

Fig. 2. Snippet example

C. Matching

The matching process between a StackOverflow post and a code snippet relies on shared keywords to establish contextually relevant connections. By leveraging tags, extracted keywords, and JSDoc annotations within code, we can create a straightforward yet effective system for linking questions and answers with real-world examples. This keyword-based approach focuses on matching common programming terms, specific functions, or general problem descriptions between the content of StackOverflow posts and annotated snippets. Through this method, relevant StackOverflow discussions become more accessible for developers seeking practical, applicable solutions directly tied to the issues they are working on.

When processing a code snippet, the system identifies keywords embedded within the annotations, such as @tags and @description, which outline the snippet's purpose, functions, or applicable technology stacks. Similarly, keywords from StackOverflow questions, answers, and associated tags are indexed and used to filter relevant discussions. This enables a comparison of terms in both sources, highlighting instances where the language, technology, or functionality in a StackOverflow post aligns with the snippet's description. For example, if a code snippet is tagged with keywords like "clipboard" and "JavaScript," posts on StackOverflow discussing clipboard operations in JavaScript will be prioritized, as they offer the most applicable insights.

By implementing a keyword-matching strategy, the system builds relationships based on relevance, allowing developers to locate not only code that solves similar problems but also examples suited to their specific language or platform requirements. As this matching process becomes more sophisticated, it opens up possibilities for further refinement, such as

weighting keywords according to their relevance or frequency. This adaptable model lays the groundwork for more advanced features in the future, including natural language processing techniques that could improve the precision and accuracy of the matching system.

D. Neo4j

Neo4j is a graph database management system designed to handle large-scale and highly connected datasets, and therefore will be the database of choice for this project. By using a graph-based model, Neo4j allows us to represent and query complex relationships between StackOverflow posts and code snippets in a more efficient and visually more didactic way [10]. In this context, each StackOverflow post, code snippet, tags, file extensions, and operating systems become entities that relate to each other; in figure 3 you can see an example of a relationship between these entities.

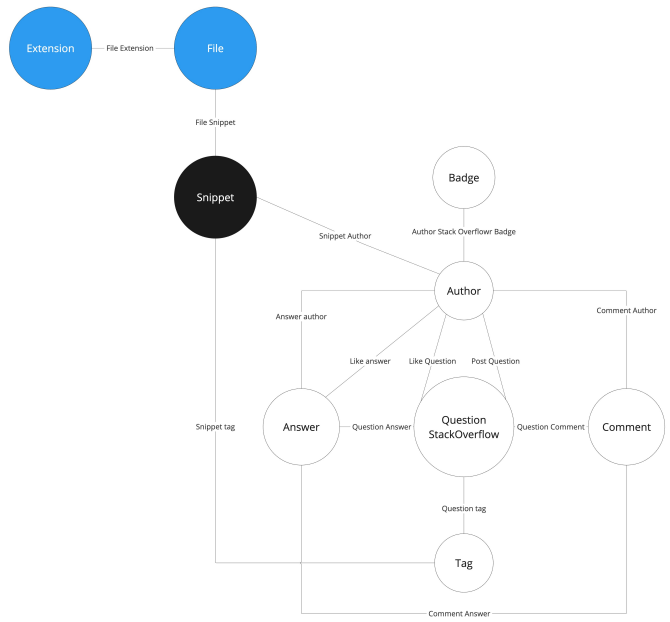

Fig. 3. Relationship between Posts and Snippet

The tags were used to connect the snippets to the StackOverflow posts. This structure allows us to create a relationship between N posts and N snippets. Several important insights can be observed, including the goal of this work, the relationship between the snippets and StackOverflow. In the future, work could be done to identify better solutions according to a specific operating system and/or file extension, allowing developers to quickly access relevant content that would otherwise require time-consuming searches across multiple platforms.

E. Google Extension

A Google Chrome extension is a small software program that enhances browser functionality by allowing developers to modify and interact with web pages in real time. Extensions operate within the browser and can access specific websites or URLs based on predefined permissions, enabling custom modifications to the user's browsing experience. They consist of HTML, JavaScript, and CSS files, and are defined by a manifest file that outlines permissions, scripts, and rules for when and where the extension should operate. This framework allows developers to manipulate the Document Object Model (DOM) of a webpage directly, which can alter the visual structure, insert interactive elements, or retrieve data dynamically from external sources.

This project will use a Chrome extension to interact with the DOM of StackOverflow pages, adding custom elements that display related code snippets based on the tags of the current post. By analyzing the tags associated with each question, the extension will query an external database for relevant code snippets and other StackOverflow posts that share similar tags. These snippets will be presented in a fixed overlay on the page, providing the user with immediate access to contextually relevant code that addresses similar problems. This integration enhances the StackOverflow experience by offering practical, real-world code examples and connections to related questions, thereby aiding developers in finding comprehensive solutions without needing to leave the StackOverflow page.

F. Architecture

The architecture is designed to seamlessly integrate with StackOverflow, providing users with an enhanced experience through relevant code snippets and related posts. As illustrated in Figure 4, the system consists of four main components: the user interacting with StackOverflow, the Sahub Chrome extension, Sahub Application (frontend) and the Sahub server, which is backed by a Neo4j database.

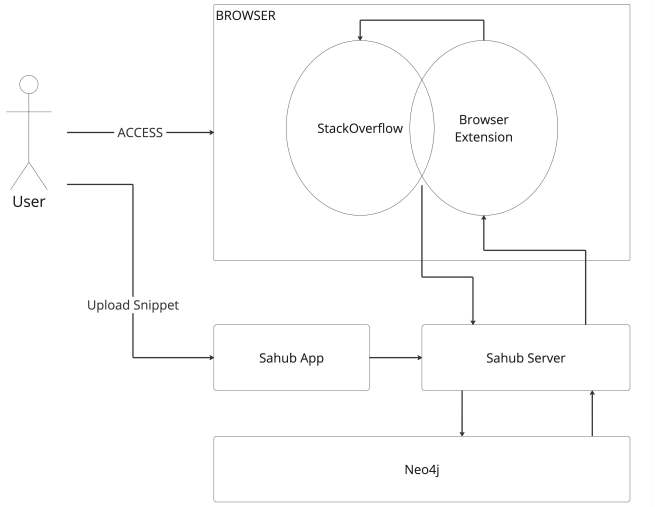

Fig. 4. Sahub Archutecture

When a user accesses a question on StackOverflow through their browser, the Sahub Chrome extension is triggered. The extension captures the metadata of the current post, such as its tags, title, and URL, and sends this information to the

Sahub server. This process ensures that the post is stored in the Neo4j database, enabling future queries and relationships to be established. Simultaneously, the extension queries the Sahub server for related snippets and posts, based on the tags of the current question. Once the results are retrieved, the extension dynamically injects a sidebar into the DOM of the StackOverflow page, displaying the relevant snippets and posts directly to the user. This can be seen in the sequence diagram in the image 5.

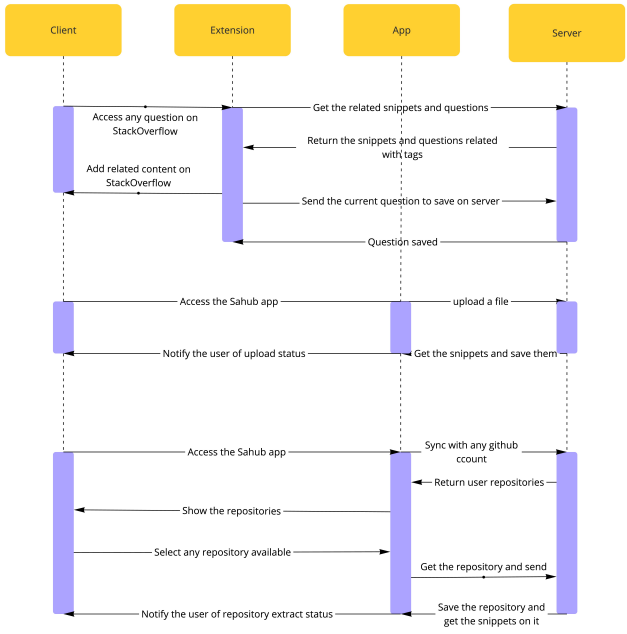

Fig. 5. Sequence diagram

To add new Snippets the users can access a simple Sahub web application. This platform allows users to upload their own snippets or connect in a github repository to get automatically, which are enriched with metadata such as tags, descriptions, authorship, and other details. These snippets are then processed by the Sahub server and stored in the Neo4j database, making them accessible to other users via the extension.

IV. Results

A. Database data relationship

The results obtained can be seen by visualizing Figure 6, where StackOverflow posts and code snippets are linked using shared tags. This representation provides a practical example of how the knowledge graph enables intuitive navigation between questions and relevant code examples based on common keywords.

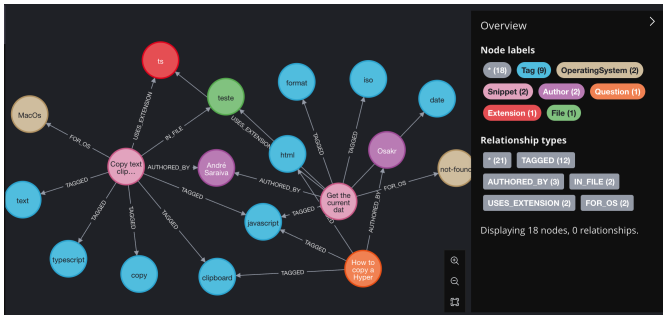

Fig. 6. Graph result

In the figure, each node represents a specific entity, with distinct colors for each type. For example, the StackOverflow post titled ”How to copy a HyperText link into clipboard without losing the link properties” is associated with tags such as ”clipboard” and ”JavaScript”. Meanwhile, the code snippet node labeled ”Copy text to clipboard” is tagged with similar keywords such as ”clipboard” and ”JavaScript,” as well as additional metadata such as the file extension (ts) and operating system (MacOS).

This alignment of tags allows the system to create a direct relationship between the question and the code example, as shown by the lines connecting the shared keywords. The visual model also highlights the role of additional metadata, such as @fileExtension and @operatingSystem, which provide more specificity to the search and retrieval process. In this case, developers searching for clipboard-related functions in JavaScript will not only find relevant code snippets, but will also be able to filter results based on operating system compatibility and file type. This targeted search capability makes the knowledge graph a powerful tool for retrieving highly specific information, facilitating faster problem resolution and increasing developer productivity.

B. Google Extension

To test the relationship between StackOverflow questions and code snippets, a Google Chrome extension was developed to display related snippets directly on the StackOverflow question page. In Figure 7, you can see the sidebar added by the extension, which provides a convenient overview of related snippets and other relevant posts. This sidebar enhances the user experience by offering immediate access to code examples and additional questions that share similar tags, making it easier for developers to find contextually relevant solutions without navigating away from the page.

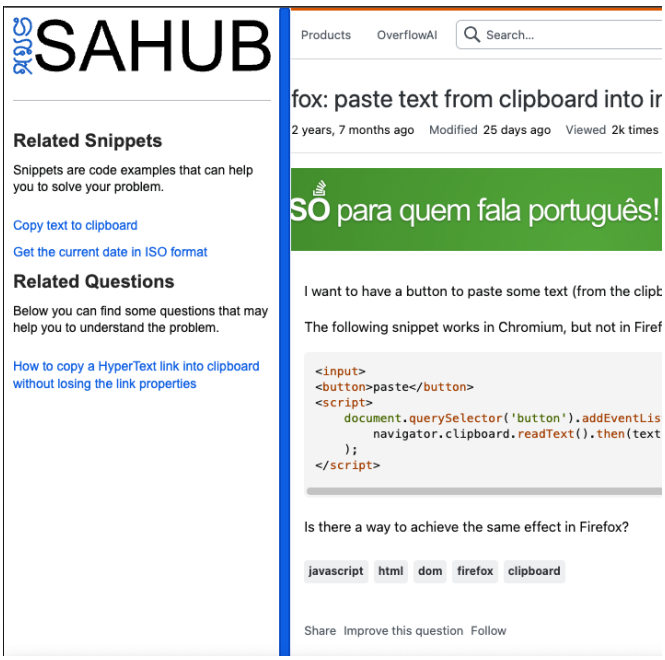

Fig. 7. Sahub sidebar on StackOverflow

All snippets appear in a dedicated section created by the extension below the main question. In this section, users can view the code, author name, description, and associated tags for each snippet. If a user clicks on a snippet, they are taken directly to the full code or source link, allowing them to explore the snippet in its original context.

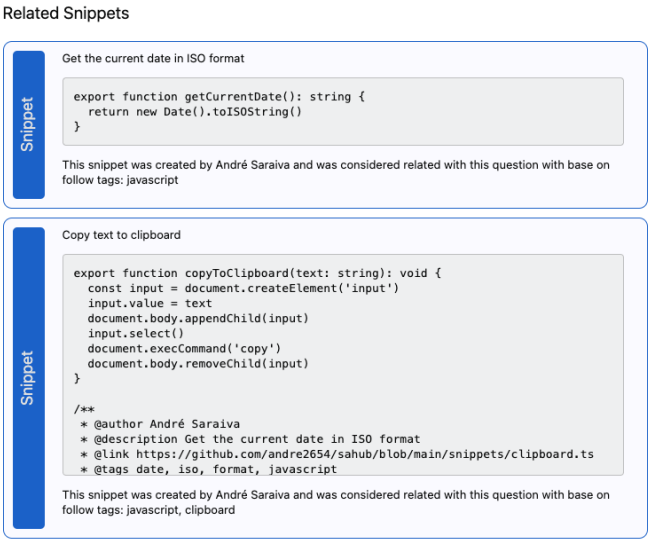

Fig. 8. Sahub snippets on StackOverflow

This ”Snippets” section, shown in Figure 8, makes it simple for users to access practical code examples and reference other related topics, streamlining the problem-solving process on StackOverflow.

C. Sahub Application

Sahub Application The Sahub application provides a user-friendly interface that allows developers to submit their code snippets and integrate them into the Sahub ecosystem. As shown in Figure 9, users have two primary options for uploading their snippets. First, they can manually submit individual files containing snippets through the ”Upload” functionality. This feature enables the application to parse the file, identify snippets using the predefined annotation syntax (e.g., @description, @tags, @author), and automatically upload them to the Neo4j database. This ensures that the submitted snippets are enriched with metadata and can be later linked to relevant StackOverflow posts.

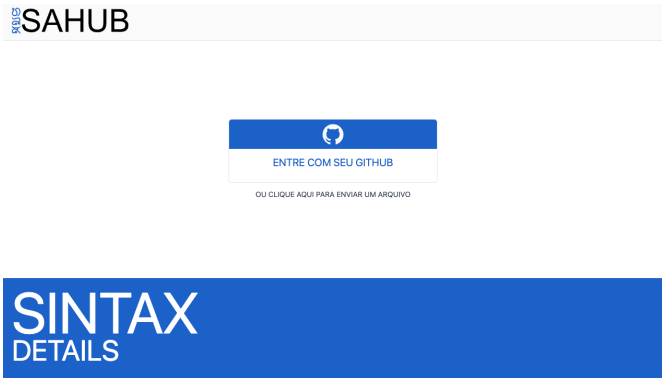

Fig. 9. Sahub application

In addition to manual uploads, the Sahub application offers seamless integration with GitHub. By logging in with their GitHub accounts, users can browse their repositories and select a specific project from which snippets will be extracted. Once a repository is selected, the application automatically scans all files, identifies snippets based on the annotation syntax, and processes them for storage in the database.

V. Conclusions

This project establishes a foundational model for connecting StackOverflow posts with real-world code snippets through a graph-based approach using Neo4j. By leveraging shared tags and JSDoc annotations, a structured and interactive graph was created that allows developers to access contextually relevant code examples directly linked to community-generated questions. This keyword-based matching strategy, while simple, has proven effective in associating queries with practical solution approaches, thereby increasing the usability and relevance of StackOverflow as a programming resource.

In addition, this graph model demonstrates the potential for scalability and adaptability. By integrating the StackOverflow API, the system can be continuously updated, ensuring that the graphs reflect the latest trends and challenges in software development. With this initial model, it is possible to lay the foundation for a knowledge base that can serve as a reference for developers, simplifying the process of finding contextually applicable code within the vast repository of StackOverflow posts.

A. Future directions

This approach establishes a solid foundation for future enhancements, including the integration of machine learning techniques to refine and expand the matching process. By analyzing patterns within the relationships between StackOverflow posts and code snippets, it would be possible to develop predictive models capable of anticipating relevant solutions based on specific developer queries. This could transform StackOverflow into a more responsive, context-aware resource, providing developers with targeted code suggestions tailored to their needs.

The current graph structure, with its clearly defined and interconnected relationships, represents an essential first step toward creating a robust and contextually relevant knowledge base. While the integration with GitHub is already functional, enhancing this connection to support automatic updates from repositories would significantly improve the system. By tracking changes in connected repositories, such as modified or newly added files, the system could automatically extract updated snippets and synchronize them with the knowledge base. This would ensure that the database remains up-to-date with the latest code changes, providing users with more accurate and timely examples, while reducing the need for manual intervention.

ACKNOWLEDGMENT

The authors are grateful to the Foundation for Science and Technology (FCT, Portugal) for financial support through national funds FCT/MCTES (PIDDAC) to CeDRI (UIDB/05757/2020 and UIDP/05757/2020) and SusTEC (LA/P/0007/2021).

REFERENCES

[1] S. M. Nasehi, J. Sillito, F. Maurer, and C. Burns, "What makes a good code example?: A study of programming q&a in stackoverflow," in *2012 28th IEEE International Conference on Software Maintenance (ICSM)*, 2012, pp. 25–34.

[2] B. Vasilescu, V. Filkov, and A. Serebrenik, "Stackoverflow and github: Associations between software development and crowdsourced knowledge," in *2013 International Conference on Social Computing*, 2013, pp. 188–195.

[3] S. Wang, D. Lo, and L. Jiang, "An empirical study on developer interactions in stackoverflow," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, ser. SAC '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 1019–1024. [Online]. Available: https://doi.org/10.1145/2480362.2480557

[4] A. Ramirez-Lopez and D. Muñoz, "Increasing practical lessons and inclusion of applied examples to motivate university students during programming courses," *Procedia - Social and Behavioral Sciences*, vol. 176, pp. 552–564, 2015, international Educational Technology Conference, IETC 2014, 3-5 September 2014, Chicago, IL, USA. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877042815005479

[5] M. A. Musen, "Dimensions of knowledge sharing and reuse," *Computers and Biomedical Research*, vol. 25, no. 5, pp. 435–467, 1992. [Online]. Available: https://www.sciencedirect.com/science/article/pii/001048099290003S

[6] M. Asaduzzaman, A. S. Mashiyat, C. K. Roy, and K. A. Schneider, "Answering questions about unanswered questions of stack overflow," in *2013 10th Working Conference on Mining Software Repositories (MSR)*. IEEE, 2013, pp. 97–100.

[7] L. Ponzanelli, A. Bacchelli, and M. Lanza, "Seahawk: Stack overflow in the ide," in *2013 35th International Conference on Software Engineering (ICSE)*, 2013, pp. 1295–1298.

[8] L. Ponzanelli, G. Bavota, M. Di Penta, R. Oliveto, and M. Lanza, "Mining stackoverflow to turn the ide into a self-confident programming prompter," in *Proceedings of the 11th Working Conference on Mining Software Repositories*, ser. MSR 2014. New York, NY, USA: Association for Computing Machinery, 2014, p. 102–111. [Online]. Available: https://doi.org/10.1145/2597073.2597077

[9] S. Gottipati, D. Lo, and J. Jiang, "Finding relevant answers in software forums," in *2011 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011)*. IEEE, 2011, pp. 323–332.

[10] J. J. Miller, "Graph database applications and concepts with neo4j," in *Proceedings of the southern association for information systems conference, Atlanta, GA, USA*, vol. 2324, no. 36, 2013, pp. 141–147.