IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.53

Volume 4, Issue 2, January 2024

Code Buddy

Ms. Akansha Bibishan Zambare[1], Ms. Vaishnavi Ghananil Patil[2],
Ms. Sayali Mahesh Nikam[3], Mrs. Swati Patil[4]
Students, Department of Computer Technology[1,2,3]
Lecturer, Department of Computer Technology[4]
Bharati Vidyapeeth Institute of Technology, Navi Mumbai, Maharashtra, India

Abstract*: Software development is inherently linked to the challenge of finding and fixing bugs in code. This research explores alternative methods of code debugging and debugging, aimed at increasing code quality, reducing development time, reducing software failures We provide a multi-faceted, AI-powered bug-finding project detection, and proactive prevention strategies are providing.*

Keywords: Code error detection, static analysis, dynamic analysis, intelligent assistance, proactive prevention, software quality, developer productivity

I. INTRODUCTION

In the intricate world of software development, the quest for excellence and efficiency remains an ever-ongoing pursuit. The success of software projects depends on the capacity to craft code that is not only functional but also resilient, maintainable, and free of errors. As software developers, we navigate a labyrinthine landscape of code structures, libraries, and dependencies, where even the slightest oversight can lead to formidable challenges.

Code Buddy is not a mere addition to the toolbox of developer tools; it represents a comprehensive solution, a web-based platform that sets out to tackle the fundamental challenges encountered in software development. It's more than just a tool; it's a virtual companion for developers, offering a triumvirate of solutions that promise to redefine our approach to coding.

One of the foundational pillars of Code Buddy's capabilities is its error detection feature. Software errors, from routine syntax issues to elusive logical defects, have long vexed developers. Code debugging, an intrinsic aspect of software development, is often an intricate and painstaking task. It is the stage where the minutiae make all the difference between a functional program and an impenetrable enigma. Here, Code Buddy offers a lifeline to developers through an array of debugging features. It empowers developers to traverse their code, establish breakpoints, and scrutinize variable values, all while providing real-time feedback and actionable suggestions.

Code documentation, often relegated to an afterthought, is a pivotal component of code clarity and collaboration. It serves as the roadmap guiding developers and acts as a means of communication among team members. Code Buddy recognizes this necessity and offers automated comment generation tools designed to create coherent, well structured comments for code. Through this approach, Code Buddy enhances code readability and maintainability, making documentation an integral part of the development process.

II. LITERATURE REVIEW

Ribeiro, Marco Tulio. "Exploring Intelligent Artificial Intelligence (XAI) Techniques for Deep Learning in Software Engineering." arXiv Preprinted arXiv:2002.08405 (2020). This paper provides a detailed description of XAI techniques such as focus, special pressure maps, and counterfactual reasoning. Adjustments can be made in the following ways: Emphasize important pieces of the code: Think machines look at the parts of the code that get the most attention from the prototype as they are developed, and can determine key parts of the argument Create a heat diagram: Basic stress diagrams visually represent the "concept" of the model in the code, and identify areas that contribute most to the outcome or decision-making process Sketch alternative scenarios: Counterfactual reasoning can explore "what if" scenarios by looking at application of different aspects of the law and changes in outcomes, helping to explain the impact of specific pieces of the law get inside

DOI: 10.48175/568

Iyer, Shankar, et al. "Code2Nl: Provides natural language descriptions of code." Proceedings of the 2018 Conference on Methods in Natural Language Processing. 2018. This work proposes Code2Nl, a deep learning model that provides a summary of Python code in natural language. If you extend Code2Nl you can: Define roles and classes: Create a human-readable summary of their goals and actions, and increase understanding for less experienced entrepreneurs. Paraphrase complex logic: Translate complex code blocks into simple language, help with troubleshooting and code analysis. Write code with natural language: Use summaries to write the code's actions and intentions instead of manually entering text.

III. PROPOSED METHODOLOGY

3.1 Features of Code Buddy

- Code Error Detection
- Static analysis methods for detecting common design errors.
- Dynamic analysis methods for catching runtime errors.
- Popular interfaces and IDEs for real-time feedback. Help identify security vulnerabilities.

A. Translate code

- Automatic syntax and semantic interpretation. a specific language for products and content management.
- Address differences in libraries and APIs.

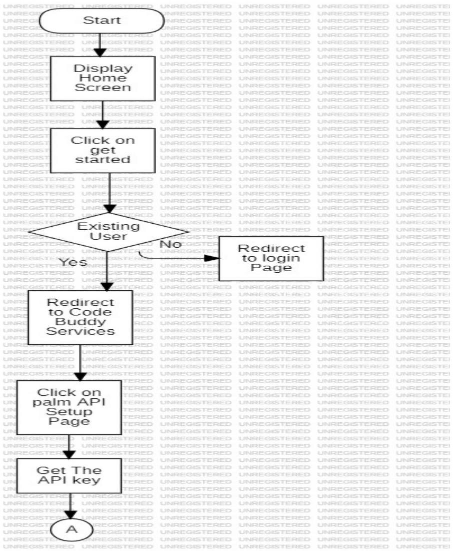
B. Test Case Generation:

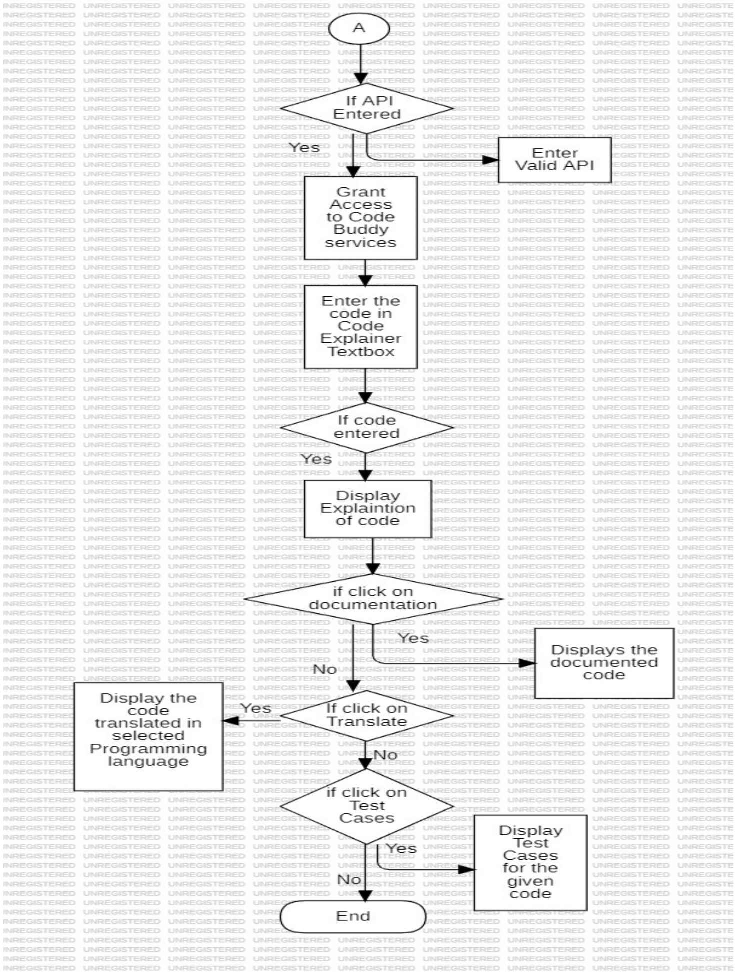
- Automated generation of test cases.
- Adaptive test methods based on complex rules.
- Integration of regulatory overlay tools.
- Support for positive and negative test results.

C. Auto Documentation:

- Automatically generate and maintain documentation in sync with code changes.
- Use natural language generation to create clear and concise explanations.
- Integrate with diverse coding styles and existing tools.

3.2 Flowchart




A
If API Entered
Yes
Enter Valid API
Grant Access to Code Buddy services
Enter the code in Code Explainer Textbox
If code entered
Yes
Display Explaintion of code
if click on documentation
Yes
Displays the documented code
No
Display the code translated in selected Programming language
Yes
If click on Translate
No
if click on Test Cases
Yes
Display Test Cases for the given code
No
End

IV. CONCLUSION

The literature review highlights the promising prospects and challenges associated with the Code Buddy website, a platform envisioned to redefine collaborative coding, foster peer learning, and cultivate a supportive programming community. As online collaboration becomes increasingly integral to the software development landscape, the Code Buddy concept offers a unique blend of interactive coding experiences and social learning principles.

The exploration of existing collaborative coding platforms underscores the need for the Code Buddy website to differentiate itself by providing a more learner-friendly environment. The emphasis on peer learning and skill development aligns with research suggesting that collaborative efforts significantly contribute to accelerating the learning curve for programmers across various skill levels.

Our Code Debugging Website continues to be an asset within the programming community as it empowers developers to create code that's both reliable and efficient. We sincerely appreciate the support we have received and remain committed, to development that will positively impact programmers worldwide.

Community building emerges as a crucial factor, as social interactions within coding communities positively impact motivation and engagement. The success of the Code Buddy website hinges on creating a space where programmers feel connected, share experiences, and collaboratively overcome coding challenges.

With thoughtful consideration of the challenges and opportunities outlined in the literature, the development and implementation of a Code Buddy website have the potential to reshape online programming education, providing a dynamic and supportive space for programmers to learn, collaborate, and thrive.

REFERENCES

[1] A Survey of Static Code Analysis for Developers by Matteo Morandini, Andrea Pastorelli, and Davide Ronzitti. ACM Computing Surveys, vol. 54, no. 6, pp. 1-40, Aug. 2021.

[2] Generating Natural Language API Documentation from Source Code by Steven P. Reiss. IEEE Transactions on Software Engineering, vol. 29, no. 3, pp. 220-236, Mar. 2003.

[3] AI-Driven Static Code Analysis for Python by Eric Bodden, Amir Hosseini, Ricardo Martinez-Peinado, Andreas Zeller, and Martin Weißkopf. Proceedings of the 2020 ACM SIGSOFT International Symposium on Foundations of Software Engineering, pp. 852-867, 2020.

[4] W3school:-https://www.w3schools.com/python/matplotlib_labels.asp

[5] Tutorialpoint:-https://www.tutorialspoint.com/python/index.html

[6] GeeksforGeeks:- https://www.geeksforgeeks.org/advanced-python-tutorials/?ref=lbp

DOI: 10.48175/568