IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

CODE SENSEI – A.I CODING ASSISTANT

[1]Rayyan Azmi, [2]Vaibhav Bankar, [3]Sandesh Yadav, [4]Mrs Manjiri Gogate

[1]Student, [2]Student, [3]Student, [4]Assistant Professor

[1]Department Of Electronics and Computer Science,

[1]Shree L R Tiwari College of Engineering, Mira Road, India

Abstract: Programmers face situations where they have to rely on messy documentation, other developers and online search for basic programming commands and queries when they encounter any new programming environment. This leads to the waste of time of developers and decreases productivity. Code Sensei is a revolutionary tool aimed at streamlining the coding process for developers of all levels. By leveraging advanced machine learning algorithms, it offers real-time code suggestions, analysis, and generation. This enhances productivity, reduces coding errors, and ultimately saves valuable development time & which assists the programmers with basic programming queries that they face when they are new to a programming environment. It allows students to undertake programming exercises and receive interactive guidance in getting their programs to compile and run. This is the tool developed to assist students and instructors in solving their practical lab session problems, which comes with features such as error handling and solve common problems related to coding.

Keywords: "Artificial Intelligence", "Assistant", "Online learning", "Generative model", "Encoder-Decoder".

I. INTRODUCTION

The "Code-Sensei" will be able to develop an intelligent software tool that empowers developers by providing real- time, contextually accurate, and personalized coding guidance. By leveraging artificial intelligence and natural language processing, the project aims to enhance developer productivity, streamline coding workflows, and foster a collaborative and supportive coding environment.

Artificial Intelligence: The Code Sensei lies in the power of Artificial Intelligence (AI), transforming the way developers work.

By harnessing AI algorithms, Code Sensei analyzes intricate coding problems swiftly and accurately, ensuring developers receive instant guidelines tailored to their queries. This real- time support system not only accelerates troubleshooting but also enhances learning by providing immediate insights into coding challenges. Through AI, Code Sensei becomes an intelligent mentor, guiding developers through complexities while imparting invaluable knowledge. This fusion of AI technology and instant guidelines not only streamlines development processes but also nurtures a generation of adaptive, efficient, and highly skilled programmers, shaping the future of software engineering.

Personalized Assistance and Quality Assurance: Code Sensei stands out as a pioneering project by seamlessly integrating personalized assistance and quality assurance in software development. Tailoring its guidance to individual developer needs, Code Sensei not only boosts coding proficiency but also ensures each programmer's unique strengths are nurtured. Simultaneously, it acts as a vigilant guardian of code quality. By providing real-time feedback, it acts as a meticulous quality assurance partner, identifying errors and refining code structures. This dual functionality not only enhances developers' skills but also guarantees the delivery of high-quality, robust applications. In the dynamic landscape of software creation, Code Sensei emerges as a reliable companion, nurturing talents while maintaining impeccable coding standards.

Objective: The primary objective of the "Code Sensei" project is to revolutionize the software development landscape by offering personalized, real-time coding assistance. By leveraging cutting-edge artificial intelligence, the project aims to empower developers with instant, context-specific guidelines, enhancing their efficiency, boosting confidence, and ensuring the production of high-quality code. Through this initiative, we aspire to create a nurturing environment where developers of all levels can flourish, making software development more accessible, collaborative, and innovative.

II. LITERATURE SURVEY

Computer programming requires correct understanding of programming languages, concepts and problem-solving skills. According to a survey done by Lahtinen et al. (2005), most difficult issues that the programming students face are due to the lack of understanding of how to design a program to solve a certain task and to find bugs from their own programs. Therefore, beginners need greater practice and more learning materials to develop their programming skills [1]. The popular method of providing greater practice to students is to conduct practical programming labs.[3] Large language models (LLMs) power a rapidly increasing number of applications, having reached a proficiency in natural language that allows them to be commanded and prompted to perform a variety of tasks (OpenAI,2023). By utilizing large, in- domain datasets, their efficacy can be greatly improved for applications that require a combination of both 0natural and domain- specific language and understanding of specialized terminology. By training on domain-specific datasets, they have proved effective more broadly on applications that require advanced natural language understanding. A prominent use-case is the formal interaction with computer systems, such as program synthesis from natural language specifications, code completion, debugging, and generating documentation [3]. The custom-made dataset creation requires the expertise of a person who knows about the python development environment. This dataset was created by our team members. The other two datasets are extracted from a data dump of question-answering websites and technical forums on the internet. The detailed descriptions about these datasets are as follows. Technical sub-reddits dataset: Reddit is a discussion platform where people can discuss a topic under a forum called sub-reddit. A user can ask a question in a post and the person who knows the answer can write the answer to that question. Each post has a comments section where a person can ask clarifying questions about the post or about the topic. We used Python Reddit API Wrapper (PRAW) to extract comments and the replies to those comments to develop a question-answer pairs dataset. The sub-reddits, Python and Ubuntu, were considered to extract data of two months.

Stack Overflow and Stack Exchange dataset: Stack Overflow and Stack Exchange are question-answering websites. Stack Overflow is extensively used by developers to gather specific information about a problem faced during programming. Stack Exchange is the same as Stack Overflow but it has different domains under which people can ask questions relevant to that domain. In the case of Stack Overflow, question- answer pairs related to python and ubuntu tags are collected. In case of Stack Exchange, question-answer pairs related to relevant domains, ask ubuntu, unix and superuser, is collected.[2]

III. FLOWCHART

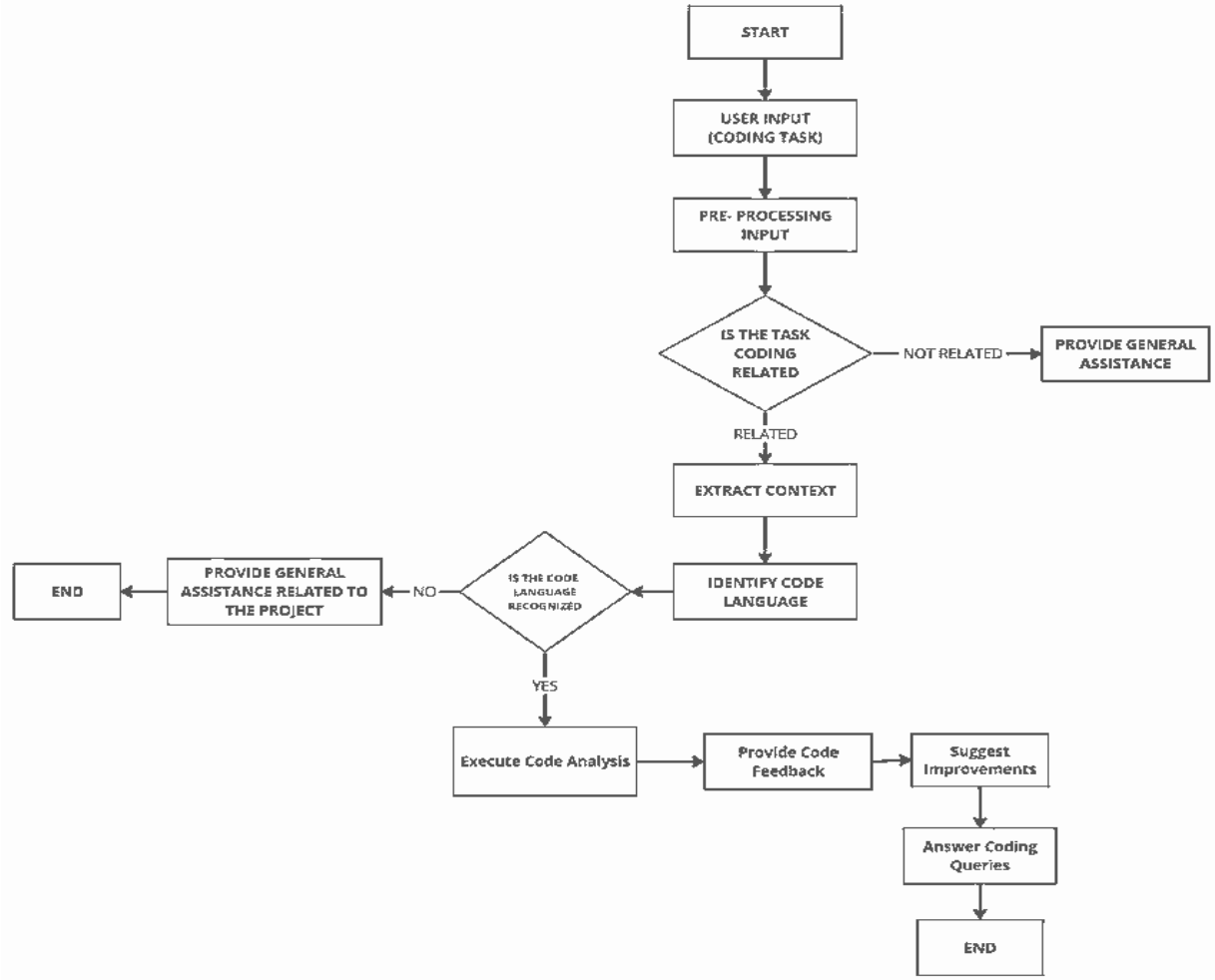

Figure 1. Flowchart.

1. "Start" symbol, representing the start of the process.
2. The user inputs a coding task into the AI coding assistant.
3. The input goes through a pre-processing step, which might involve cleaning, tokenizing, or normalizing the user's input.
4. The AI coding assistant determines whether the task is related to coding or not.
5. If the task is not code-related, the assistant provides general assistance, such as answering non-coding queries or providing general information.
6. If the task is code-related, the assistant proceeds to extract keywords or context from the input to better understand the user's intent.
7. The AI assistant identifies the code language used in the task.
8. If the code language is recognized, the assistant executes code analysis to understand the structure and logic of the code.
9. Based on the code analysis, the assistant provides code feedback, suggesting improvements and answering coding queries.
10. Finally, the flowchart ends with the "End" symbol indicating the completion of the process.

IV. SOFTWARE REQUIREMENTS

1) **Python Compiler**: Python is a high-level, general-purpose programming language. It is used for developing websites and software, task automation, data analysis, and data visualization.
2) **Visual Studio Code**: Visual Studio Code is a code editor redefined and optimized for building and debugging modern web and cloud applications.
3) **Google Collaboratory**: Collab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.

V. METHODOLOGY

Understanding User Requirements and Research: The methodology begins with a focus on understanding user needs through interviews and surveys with developers. This feedback guides the project's scope and objectives. A thorough literature review explores existing AI coding solutions and the latest in AI and NLP techniques. This dual approach ensures the project aligns with user expectations while leveraging cutting-edge technologies.

Data Collection, Model Development, and Integration: The next phase involves collecting diverse code repositories
for training the AI models. After data preprocessing, the selection of suitable algorithms and frameworks leads to the development and training of the models. Integration with popular IDEs and code editors ensures accessibility, and the creation of a user-friendly interface enhances the coding experience. This phase prioritizes the seamless integration of the AI coding assistant into developers' workflows.

Feedback Mechanism, Iterative Development, and Testing: The methodology emphasizes user involvement through feedback mechanisms and iterative development. Early feedback informs the continuous refinement of AI models, algorithms, and the user interface. Rigorous testing, including unit and integration testing, ensures the correctness and effectiveness of the AI coding assistant. User acceptance testing is conducted to validate its real- world performance, fostering a collaborative development process.

Documentation, Deployment, and Continuous Improvement: Comprehensive documentation is created for developers and users, guiding them through integration and usage. Deployment on cloud platforms ensures scalability, and monitoring tools are implemented for performance tracking. Security measures, including data privacy and regular security audits, safeguard user information. The methodology concludes with a focus on continuous improvement, integrating user feedback and retraining models to adapt to evolving coding practices. This structured approach aims to deliver a reliable, effective, and user-centric AI Coding Assistant.

VI. OUTPUT

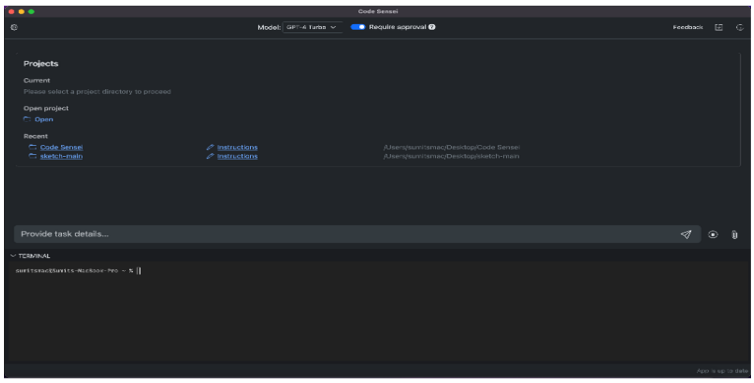

Figure 2. Home UI.

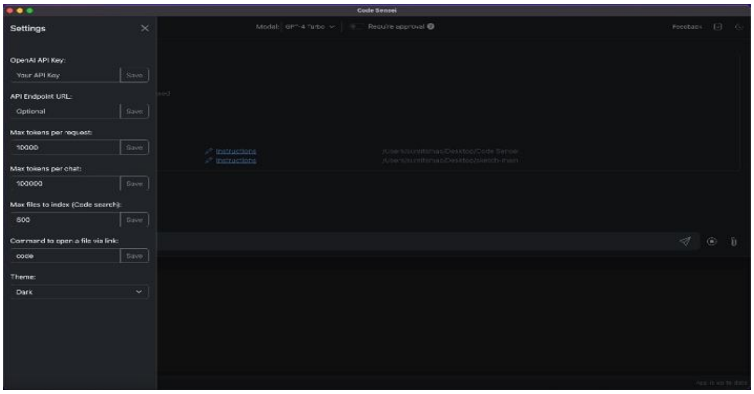

Figure 3. Setting Bar.

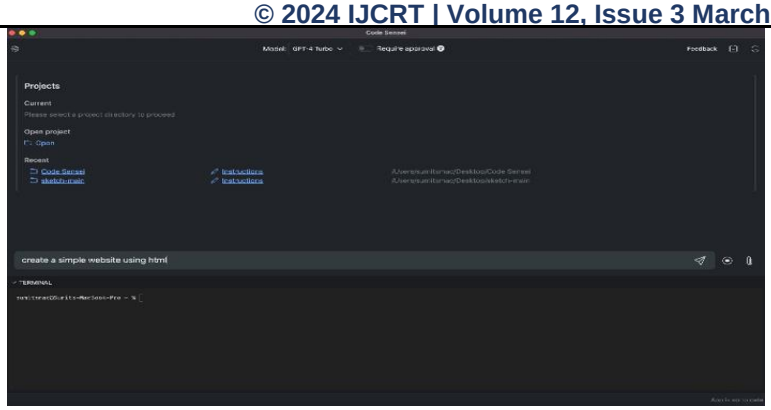

Figure 4. Input Task Detail.

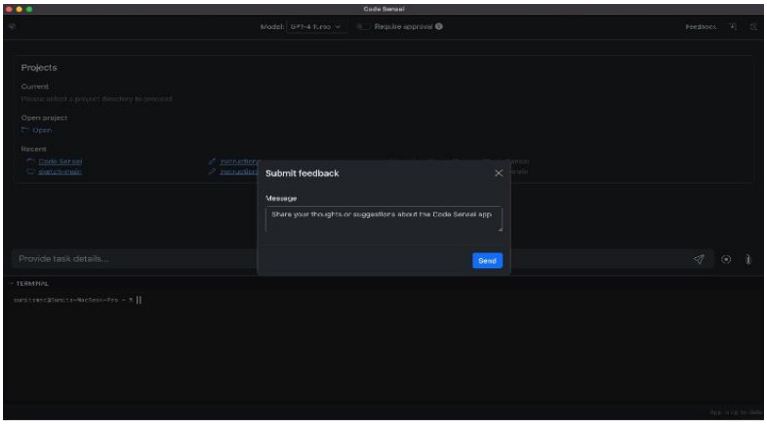

Figure 5. Feedback.

Real-time Guidance for Swift Problem-Solving: The implementation of an AI Coding Assistant promises developers instantaneous and precise coding suggestions, presenting a paradigm shift in the problem-solving landscape. Real-time guidance means that as developers engage with their code, they will receive immediate insights and recommendations, drastically reducing the time traditionally spent on troubleshooting and refining code. This agile responsiveness not only enhances the efficiency of developers but also introduces a dynamic element to the coding process, creating a more interactive and productive development environment.

Enhanced Productivity and Streamlined Workflows: The AI Coding Assistant contributes significantly to enhanced productivity by streamlining workflows and offering personalized feedback. Developers can seamlessly integrate the tool into their coding environment, receiving tailored suggestions that align with their coding styles and project requirements. The result is a more efficient coding process, allowing developers to focus on the core aspects of their work without being bogged down by repetitive problem-solving. The streamlined workflows, coupled with real-time guidance, create a synergy that accelerates project progress and efficiency.

Elevated Code Quality and Learning Enrichment: One of the notable impacts of the AI Coding Assistant is the elevation of code quality. Contextually relevant recommendations ensure that the code produced is not only functionally correct but also adheres to industry best practices and coding standards. This improvement in code quality contributes to the overall robustness and maintainability of software projects. Additionally, the AI assistant acts as a valuable learning companion, especially for novice developers. By providing enriched learning experiences through personalized feedback and real-time guidance, the tool fosters skill growth and confidence, creating a supportive environment for continuous learning and improvement.

VII. FUTURE SCOPE

The future scope of an AI Coding assistant will assist in automating repetitive tasks, optimizing code, and aiding in specific aspects of software development.

VIII. CONCLUSION

In conclusion, the AI Coding Assistant project stands groundbreaking venture in the realm of software development, revolutionizing the way developers approach their craft. By harnessing the power of artificial intelligence and natural language processing, this innovative tool has not only streamlined the coding process but has also empowered developers, from beginners to seasoned professionals, to elevate their skills and efficiency. Throughout this project, we've witnessed the transformative potential of AI coding assistants in enhancing productivity, reducing errors, and fostering a collaborative and consistent coding environment. The advantages of this technology, such as increased productivity, error reduction, skill enhancement, and improved code review processes, have been evident. However, it's essential to approach these tools with a balanced perspective, acknowledging their advantages while being mindful of potential challenges like overreliance and privacy concerns.

IX. REFERENCES

[1]. Milan Amaratunga, Milinda Deepal, Dilshan DeSilva. *"An Interactive Programming Assistance Tool (iPAT) for Instructors and Novice Programmers."* The 8th International Conference on Computer Science & Education (ICCSE 2013) April 26-28, 2013. Colombo, Sri Lanka.

[2]. Aditya Ankur Yadav, Ishan Garg, Dr. Pratistha Mathur, *"PACT - Programming Assistant Chatbot"*. 2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT) Manipal University Jaipur,
Sep28-29, 2019.

[3]. Jonas Gehring, Stan Sootla, Manish Bhatt. How to use LLM for programming tasks. *"Code Llama: Open Foundation Model for Code."* Meta AI, August 2023

[4]. Jenny T. Liang, Chenyang Yang, and Brad A. Myers. 2024. *"A Large-Scale Survey on the Usability of AI Programming Assistants: Successes and challenges".* In 2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE 2024), April 14–20, 2024, Lisbon, Portugal.