CODE GENIE

MINOR PROJECT REPORT

Submitted in partial fulfillment of the requirement for the Degree of Bachelors of Engineering in Computer Science & Engineering

Submitted To

[PARUL UNIVERSITY, VADODARA, GUJARAT (INDIA)]

Submitted By

Kishan Pokar	**2203051050791**
Dhruv Pandey	**2203051050780**
Rohan Solanki	**2203051050482**
Hans Raj	**2203051050203**

Under The Guidance of
Dr.Anand Singh Gadwal
Assistant Prof, CSE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

PARUL INSTITUTE OF TECHNOLOGY VADODARA, GUJARAT

SESSION: AY 2024-2025

Parul University
Parul Institute of Technology

(Session: 2024 -2025)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that **KISHAN POKAR, DHRUV PANDEY, ROHAN SOLANKI, HANS RAJ** Students of **CSE VI Semester** of **"Parul Institute of Technology, Vadodara"** has completed their **Minor Project** titled **"CODE GENIE"**, as per the syllabus and has submitted a satisfactory report on this project as a partial fulfillment towardsthe award of degree of **Bachelor of Technology** in **Computer Science and Engineering** under **Parul University, Vadodara, Gujarat, India**.

Dr.Anand Gadwal	**Prof. Sumitra Menaria**	**DR. Swapnil Parikh**
Asst Prof	**Head (CSE)**	**Principal**
CSE	**PIT, Vadodara**	**PIT,(PU) Vadodara**

DECLARATION

We the undersigned solemnly declare that the project report "**CODE GENIE"** is based on my own work carried out during the course of our study under the supervision of **Dr.Anand Singh Gadwal, Asst Prof, CSE**.

We assert the statements made and conclusions drawn are the outcomes of my own work. I further certify that

1. The work contained in the report is original and has been done by us under the general supervision of our supervisor.
2. The work has not been submitted to any other Institution for any other degree / diploma / certificate in this university or any other University of India or abroad.
3. We have followed the guidelines provided by the university in writing the report.

Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

Kishan Pokar	**2203051050791**	**SIGNATURE** _____
Dhruv Pandey	**2203051050780**	**SIGNATURE** _____
Rohan Solanki	**2203051050482**	**SIGNATURE** _____
Hans Raj	**2203051050203**	**SIGNATURE** _____

ACKNOWLEDGEMENT

In this semester, we have completed our project on **"CODE GENIE"**. During this time, all the group members collaboratively worked on the project and learnt about the industry standards that how projects are being developed in IT Companies. We also understood the importance of teamwork while creating a project and got to learn the new technologies on which we are going to work in near future.

We gratefully acknowledge for the assistance, cooperation guidance and clarification provided by **"Dr.Anand Singh Gadwal"** during the development of our project. We would also like to thank our Head of Department **Prof. Sumitra Menaria** and our Principal **Dr. Swapnil Parikh** Sir for giving us an opportunity to develop this project. Their continuous motivation and guidance helped us overcomethe different obstacles for completing the Project.

We perceive this as an opportunity and a big milestone in our career development. We will strive to use gained skills and knowledge in our best possible way and we will work to improve them.

Kishan Pokar **2203051050791** **SIGNATURE** _____

Dhruv Pandey **2203051050780** **SIGNATURE** _____

Rohan Solanki **2203051050482** **SIGNATURE** _____

Hans Raj **2203051050203** **SIGNATURE** _____

LIST OF FIGURES

ABSTRACT

Code Genie is a web-based application designed to simplify and automate code documentation and cross-language code conversion. It helps developers generate meaningful comments, improving code readability and maintainability while also enabling seamless translation of code between different programming languages.

The project leverages **Natural Language Processing (NLP)** and **Machine Learning (ML)** to analyze code structure, infer functionality, and generate structured documentation automatically. It also provides an intelligent translation system, allowing developers to convert code between languages like Python, Java, C++, and JavaScript with accuracy.

By automating these tedious tasks, Code Genie reduces manual effort, enhances productivity, and ensures consistent documentation standards. The platform features a user-friendly interface with real-time analysis, making it an essential tool for developers working in diverse programming environments.

This project highlights the importance of AI in software development, demonstrating how automation can improve efficiency, reduce errors, and support developers in writing high-quality, well-documented code.

INDEX

CHAPTER: I - INTRODUCTION

1.1 OVERVIEW

Code Genie is an advanced AI-powered tool that enhances the efficiency of developers by automating the tedious process of writing code documentation and facilitating cross-language code conversion. The ever-growing demand for well-documented code has driven the need for solutions that reduce manual effort while maintaining accuracy and consistency. This project aims to bridge this gap by providing an intelligent, user-friendly platform.

Documentation plays a crucial role in software development, aiding collaboration, reducing errors, and making code understandable for future developers. However, it is often overlooked due to time constraints and lack of proper tools. Code Genie addresses this issue by integrating advanced NLP models that analyze code syntax and generate structured comments.

Furthermore, the ability to convert code between languages can help developers working in multi-language environments, reducing redundant efforts. The combination of automated documentation and code conversion within a single platform sets Code Genie apart from existing solutions.

1.2 PROBLEM STATEMENTS

Many developers face persistent challenges with code documentation and language conversion due to several factors:

1. Lack of Structured Comments: Without structured comments, code becomes difficult to understand, especially for team members who did not write it. This lack of clarity often leads to increased time spent deciphering code functionality and purpose.
2. Time-Consuming Manual Translation: Manually translating code between different programming languages is not only time-consuming but also prone to errors. This process demands a significant amount of effort and expertise, diverting valuable resources from core development activities.
3. Inefficiency in Collaborative Projects: Inconsistent documentation styles across

different team members lead to inefficiencies and miscommunication in collaborative projects. This inconsistency hampers seamless knowledge transfer and can cause delays in project timelines.

Code Genie addresses these challenges by automating documentation generation and providing intelligent code translation, significantly reducing manual effort and enhancing overall code quality and maintainability. With Code Genie, developers can focus more on innovation and problem-solving, rather than getting bogged down by repetitive and error-prone tasks.

1.3 Objective of Project

- Automate **Code Documentation** – Generate meaningful comments to improve code readability and maintainability.
- **Enhance Code Readability** – Ensure structured and clear documentation for better understanding.
- **Enable Cross-Language Code Conversion** – Accurately translate code between different programming languages.
- **Reduce Manual Effort** – Minimize the time spent on writing comments and rewriting code in other languages.
- **Improve Developer Productivity** – Allow developers to focus more on coding and problem-solving rather than documentation.
- **Provide a User-Friendly Platform** – Offer an easy-to-use web interface for uploading, analysing, and converting code.
- **Ensure Accuracy and Efficiency** – Use AI and NLP models to generate high-quality documentation and translations.
- **Support Multiple Programming Languages** – Facilitate seamless code conversion for Python, Java, C++, JavaScript, etc.
- **Enhance Software Maintainability** – Help developers (for now students) easily update and manage well-documented codebases.

1.4 Applications or Scope

Applications of Code Genie

1. **Software Development** – Helps developers generate clear documentation and convert code between languages.
2. **Education & Learning** – Assists students in understanding code by providing structured comments and language translations.
3. **Code Maintenance** – Makes it easier for teams to update and manage well-documented codebases.
4. **Cross-Language Projects** – Enables seamless translation of code for multi-language development.
5. **Open-Source Contributions** – Ensures standardized documentation for collaborative coding projects.

Scope of Code Genie

1. **Automation in Coding** – Reduces manual effort in writing comments and translating code.
2. **Multi-Language Support** – Expands to support more programming languages for broader use.
3. **AI-Based Improvements** – Continuous enhancements in accuracy using advanced AI models.
4. **Integration with IDEs** – Can be integrated into coding environments for real-time documentation.
5. **Team Collaboration** – Helps teams work efficiently by ensuring consistent and readable documentation.

CHAPTER: II - LITERATURE SURVEY

2.1.Introduction

Code documentation and code conversion are essential for software development, enabling better readability, maintainability, and collaboration. However, manually writing documentation and translating code between languages is time-consuming and error-prone. Several tools and frameworks have been developed to address these challenges. Code Genie aims to improve upon these existing solutions by integrating AI-powered automation to generate structured documentation and facilitate cross-language code conversion.

This chapter explores the existing tools and techniques used for code documentation and translation, analysing their strengths, limitations, and how Code Genie differentiates itself**.**

2.2 Existing Web Applications and Tools for Code Documentation

- **1. Doxygen**
- ◈ Website: www.doxygen.nl

  ◈ Description: Doxygen is a widely used tool for generating code documentation from structured comments. It supports multiple programming languages, including C++, Java, and Python.

  ◈ Features:
- Extracts comments from source code to generate documentation in HTML, PDF, and LaTeX formats.
- Supports UML diagrams and dependency graphs.

  ◈ Limitations:
- Requires manual commenting by developers.
- No AI-powered documentation generation.

- **2. Sphinx**
- ◈ Website: www.sphinx-doc.org

  ◈ Description: Sphinx is a Python-based documentation generator commonly used for technical documentation.

  ◈ Features:
- Converts reStructuredText (reST) into various formats like HTML, LaTeX, and PDF.
- Used by Python projects like Django and NumPy.

  ◈ Limitations:
- Mainly supports Python and not other languages.
- Requires manual documentation writing.

- **3. Javadoc**
- ◈ Website: www.oracle.com/java/technologies/javase/javadoc-tool.html

  ◈ Description: Javadoc is an official Java documentation tool that generates API documentation from Java source code comments.

  ◈ Features:
- Extracts structured comments to generate HTML-based API documentation.
- Helps standardize Java documentation.

  ◈ Limitations:

- Limited to Java.
- Manual comment writing is required.

- **4. GitHub Copilot**
- ◈ Website: github.com/features/copilot
  ◈ Description: GitHub Copilot is an AI-powered coding assistant that provides real-time code suggestions and comment generation.
  ◈ Features:
- Uses OpenAI Codex to provide real-time code suggestions.
- Suggests code snippets and comments based on the context.
  ◈ Limitations:
- Focuses on code completion, not detailed documentation.
- Does not provide cross-language code translation**.**

2.3 Existing Web Applications and Tools for Code Translation

5. TransCoder (by Facebook AI Research)

◈ Website: ai.facebook.com/blog

◈ Description: TransCoder is an AI-based tool that translates code between Python, Java, and C++.

◈ Features:

- Uses deep learning models trained on GitHub datasets.
- Supports Python, Java, and C++.
  ◈ Limitations:
- Can struggle with complex logic translation.
- Not available as a web app for general users.

6. OpenAI Codex (Powering GitHub Copilot & ChatGPT Code Interpreter)

◈ Website: openai.com/research/codex

◈ Description: Codex, an AI model by OpenAI, can generate and translate code between languages.

◈ Features:

- Supports multiple programming languages.
- Can explain and translate code snippets.
  ◈ Limitations:
- Requires API integration.
- Accuracy depends on context and complexity.

7. Babel (for JavaScript)

◈ Website: babeljs.io

◈ Description: Babel is a JavaScript compiler used for converting modern JavaScript (ES6+) into older versions for browser compatibility.

◈ Features:

- Allows ES6+ to ES5 conversion.
- Plugin-based syntax transformation.
  ◈ Limitations:
- Limited to JavaScript; does not support other languages.

8. Replit Ghostwriter

- Website: replit.com/site/ghostwriter
- Description: AI-powered code assistance that offers real-time suggestions.
- Features:
  - Provides syntax correction and suggestions.
  - Supports multiple languages.
    - Limitations:
  - Focuses on autocompletion, not full translation or documentation.

CHAPTER: III - METHODOLOGY

3.1 Overview of Methodology

Code Genie follows a structured methodology to automate code documentation and enable seamless code translation across multiple programming languages. The process is broken down into the following key steps:

1. Code Parsing and Analysis

- When a user inputs code, the system **parses the syntax** to identify key elements such as variables, functions, classes, loops, and conditions.
- It **analyzes the logic and structure** of the code to understand its purpose and flow.
- The system detects **language-specific patterns** and ensures that it can correctly interpret various coding styles.

2. Automated Documentation Generation

- The tool generates **structured comments** based on the analyzed code.
- It ensures that the documentation is **contextually relevant** and provides meaningful explanations of each function and code block.
- The generated comments follow **standardized formats**, making the code more readable and maintainable.

3. Code Conversion Across Languages

- The system converts code from one programming language to another while maintaining **logical accuracy**.
- It adapts the syntax, libraries, and functions based on the **best practices** of the target language.
- The translation process ensures that the converted code is **error-free and executable** without manual modifications.

4. User Interaction and Real-Time Feedback

- The platform provides a **user-friendly web interface** where users can enter their code and receive instant documentation.
- Users can view, modify, and refine the generated comments and translations as needed.

- Real-time feedback helps improve the accuracy of documentation and provides developers with better insights into their code.

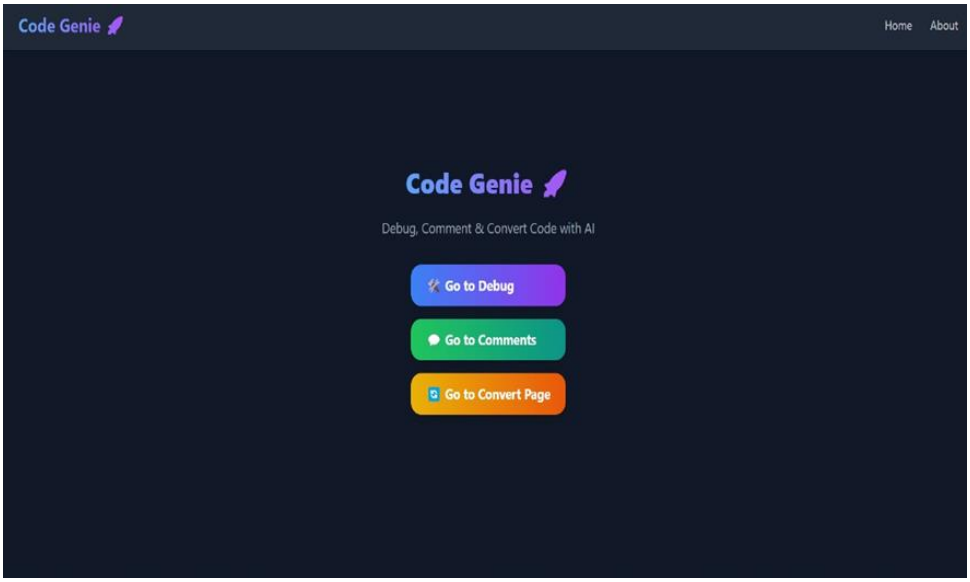

Fig 1: CODE GENIE LOGO

3.2 Project Modules

- **Frontend: React.js (for a dynamic and user-friendly UI)**
- **Backend: Node.js with Express (for handling API requests)**
  AI/NLP: Python with TensorFlow or OpenAI API (for code analysis and documentation)
- Database: PostgreSQL or MongoDB (for storing user data, projects, and code files).

3.3Diagrams

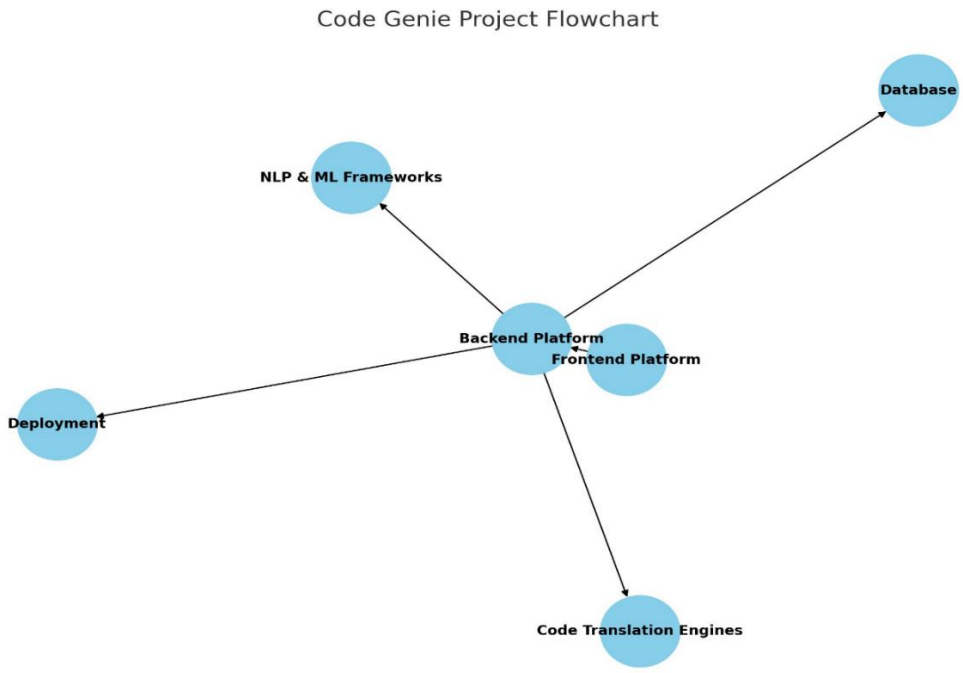

Figure 2:-Project Platforms used in Project.

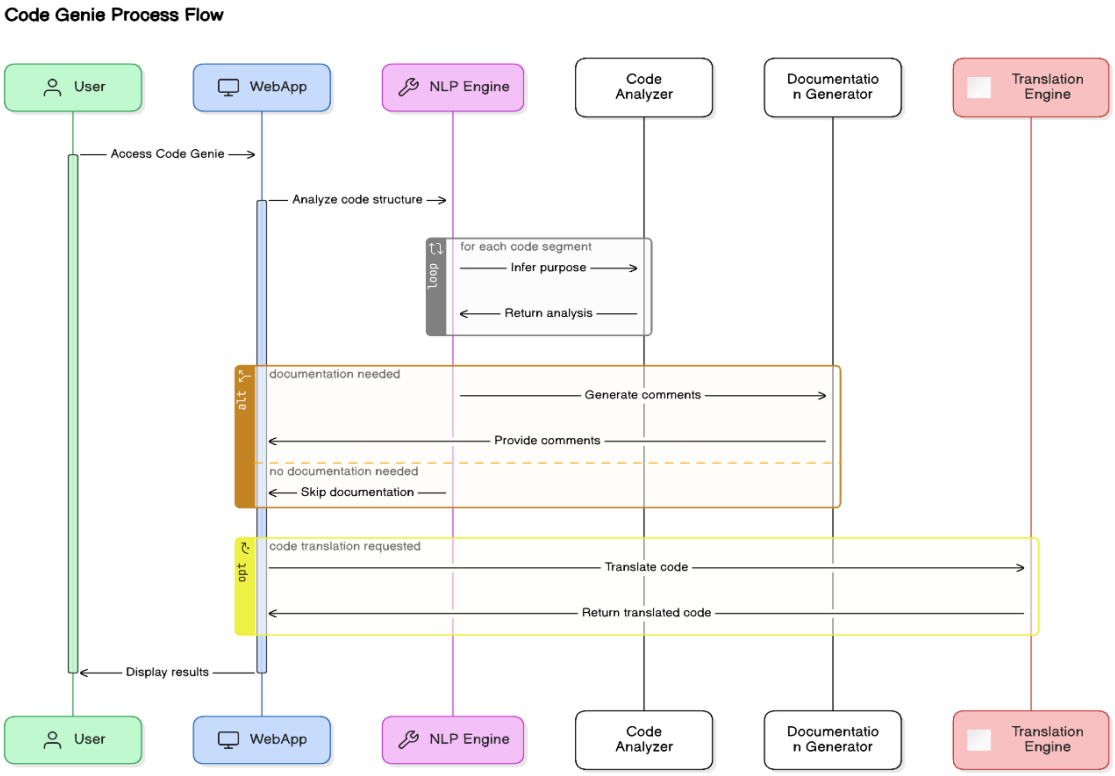

Figure 3 : ER Diagram

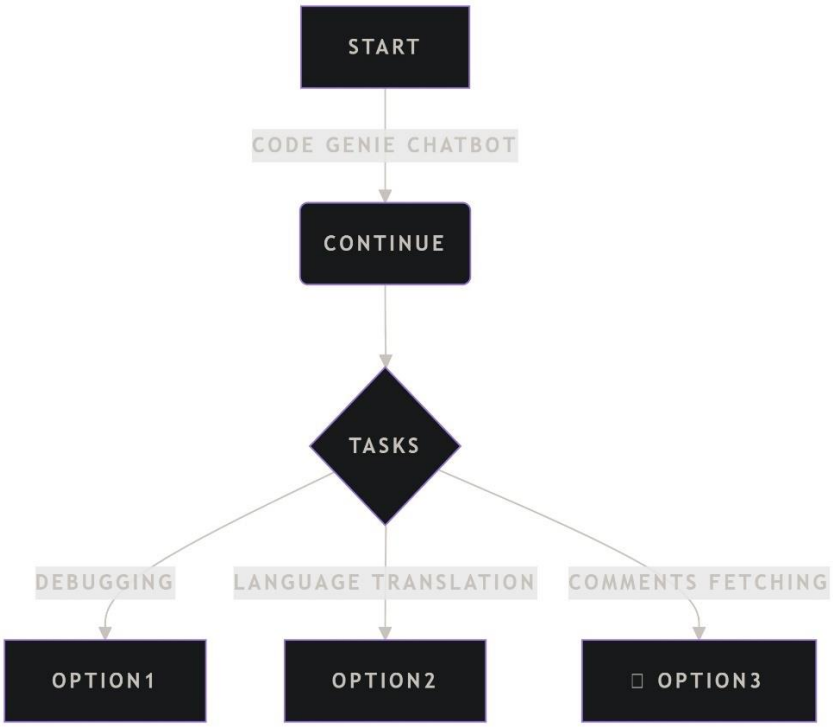

Figure 4: Flow Chart

CHAPTER: IV - SYSTEM REQUIREMENTS

4.1 Software Requirements:

- **Operating System:** Windows 10/11, macOS, Linux (Ubuntu 20.04+ recommended)
- **Backend:** Node.js (v16+) and Python (v3.8+)
- **Frontend:** React.js (with npm or yarn)
- **Database:** PostgreSQL 13+ / MongoDB 5+
- **AI/ML Libraries:** TensorFlow/PyTorch, OpenAI API, NLTK
- **Web Server:** Nginx or Apache (for deployment)
- **Cloud Services (Optional):** AWS/GCP/Azure for hosting and scalability

4.2 Hardware Requirements:

For Development & Local Testing:

- **Processor:** Intel Core i5 (8th Gen) / AMD Ryzen 5 or higher
- **RAM:** 8GB (16GB recommended for AI-based processing)
- **Storage:** 50GB SSD (100GB recommended for dataset storage)
- **GPU (Optional):** NVIDIA GTX 1650+ (for AI processing acceleration)

For Production/Server Deployment:

- **Processor:** Intel Xeon / AMD EPYC (multi-core for handling requests)
- **RAM:** 32GB+ (depending on concurrent users)
- **Storage:** 500GB SSD (scalable based on data volume)
- **GPU:** NVIDIA A100/Tesla (for AI-powered code documentation & translation)
- **Network:** High-speed internet connection for API calls and cloud integration

CHAPTER: V - EXPECTED OUTCOMES

Expected Outcomes

1. AI-Powered Code Documentation

- **Auto-Generated Comments:** Displays AI-generated comments alongside code.
- **Customization Options:** Users can modify, accept, or reject suggested documentation.
- **Syntax Highlighting:** Supports multiple languages (Python, Java, C++, JavaScript, etc.).

2. Cross-Language Code Conversion

- **Language Selection:** Dropdown menu to choose source and target languages.
- **Real-Time Translation:** Converts code instantly while preserving logic and functionality.
- **Comparison View:** Side-by-side view of original and translated code.

3. User Settings & Customization

- Change preferences like theme (dark/light mode), AI verbosity, and default programming language.

CHAPTER: VI - CONCLUSION & FUTURE SCOPE

6.1 CONCLUSION: -

In modern software development, maintaining high-quality documentation and ensuring seamless code conversion between languages are critical but time-consuming tasks. Code Genie addresses these challenges by leveraging AI-powered automation to generate meaningful documentation and facilitate cross-language code translation. By integrating Natural Language Processing (NLP) and machine learning, Code Genie enhances developer productivity and code maintainability.

1. Automating Code Documentation for Efficiency

Writing proper code documentation is often overlooked due to time constraints, resulting in unclear or inconsistent comments. Code Genie eliminates this issue by analysing code structure, inferring functionality, and generating well-structured, insightful comments. This automation not only saves valuable time but also ensures uniformity and accuracy in documentation, making the code more understandable for both individual developers and teams.

By automatically annotating functions, variables, and complex logic, Code Genie reduces the cognitive load for developers who need to comprehend unfamiliar codebases. This is particularly beneficial for large-scale projects where multiple contributors work on the same codebase, ensuring that new team members can quickly grasp the purpose of different modules without manual explanations.

2. Enhancing Code Readability and Maintainability

Readable and well-documented code is crucial for long-term maintainability. With Code Genie, developers no longer have to rely on inconsistent or missing comments. The tool provides AI-driven, context-aware explanations that describe the purpose of code snippets, making debugging, testing, and future modifications significantly easier.

Since software evolves over time, clear documentation ensures that changes can be made seamlessly without breaking functionality. Code Genie's ability to generate and refine documentation improves software quality and ensures that the code remains useful long after its initial development.

3. Facilitating Cross-Language Code Translation

Another major challenge in software engineering is the need to convert code between programming languages. Manually rewriting code to fit different language syntax and structures is error-prone and requires extensive effort. Code Genie automates language conversion by intelligently translating code

between **Python, Java, C++, and JavaScript**, ensuring accuracy while maintaining logic and efficiency.

This feature is especially useful for developers working on **multi-platform applications**, students learning new programming languages, and teams migrating projects to a different tech stack. The ability to compare original and translated code side by side helps developers validate accuracy and understand differences between programming paradigms.

4. Improving Collaboration and Productivity

With built-in project and file management, Code Genie allows teams to collaborate efficiently. Developers can easily share AI-documented code, work on translations together, and access past versions through an intuitive history tracking system. This fosters a **more streamlined workflow**, ensuring that team members have access to the latest documented code without confusion.

Additionally, Code Genie supports **exporting and sharing** annotated code, making it easy for teams to distribute documentation within their organization or educational institutions. Whether in **corporate development, open-source projects, or academic learning**, Code Genie provides a centralized and automated approach to improving code understanding.

5. Transforming Software Development with AI

By integrating AI-driven automation into documentation and language translation, Code Genie represents a **paradigm shift** in how developers interact with code. The tool reduces repetitive manual tasks, allowing developers to **focus on problem-solving and innovation** rather than spending time on writing comments or manually rewriting code.

Overall, Code Genie serves as a **powerful and intelligent assistant**, making software development more efficient, scalable, and collaborative. It not only **saves time** but also **improves code clarity, enhances team productivity, and simplifies multi-language development**, making it an essential tool for **developers, students, and software teams worldwide**.

6.2 FUTURE SCOPE: -

As an AI-driven tool for automating code documentation and translation, **Code Genie** has immense potential for further development and expansion. The future roadmap includes various enhancements to improve user experience, increase efficiency, and integrate with popular development environments. Some key areas of future scope include:

1. VS Code and IDE Extensions

- We are planning to develop an extension for VS Code and other popular Integrated Development Environments (IDEs) like JetBrains (IntelliJ, PyCharm), Eclipse, and Visual Studio.
- This extension will allow developers to access real-time AI-generated documentation and code translation directly within their coding environment.
- It will feature inline code explanations, quick language conversion, and automatic comment suggestions as developers write their code.

2. AI-Powered Code Refactoring

- Introducing an AI-based code optimization tool that suggests improvements in code structure, efficiency, and readability.
- The system will help developers identify redundant code, suggest performance improvements, and refactor legacy code for modern best practices.

3. Support for More Programming Languages

- Expanding support to additional programming languages such as Swift, Kotlin, Rust, PHP, Go, and R for broader usability.
- This will make Code Genie a universal tool for multi-language projects and software engineers working across different platforms.

4. Integration with Cloud-Based Development Platforms

- Code Genie will integrate with GitHub, GitLab, and Bitbucket to provide automated documentation and code translation within repositories.
- Cloud IDE support, such as GitHub Codespaces and AWS Cloud9, will allow remote developers to utilize AI-powered documentation seamlessly.

5. AI Chat Assistant for Code Explanation

- A built-in AI chatbot that allows developers to ask questions about specific code snippets and receive real-time explanations or suggestions.
- This feature will work as a pair-programming AI assistant, helping users understand complex logic quickly.

6. Mobile and Web-Based Code Genie App

- A lightweight web and mobile version for developers who want to quickly generate documentation or translate code on the go.
- This can be useful for **students, coding interviews, and software professionals** working remotely.

7. Collaboration & Team Features

- Adding **real-time team collaboration** where multiple developers can work on the same project and view AI-generated documentation together.
- Team-based **project sharing, annotation features, and version history tracking** will enhance productivity for organizations.

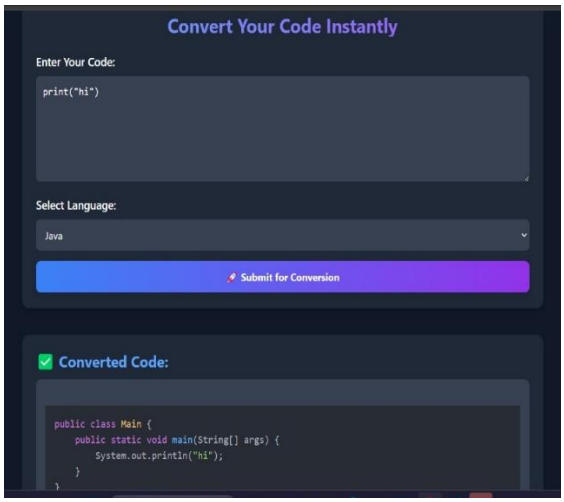

Fig. Language Conversion

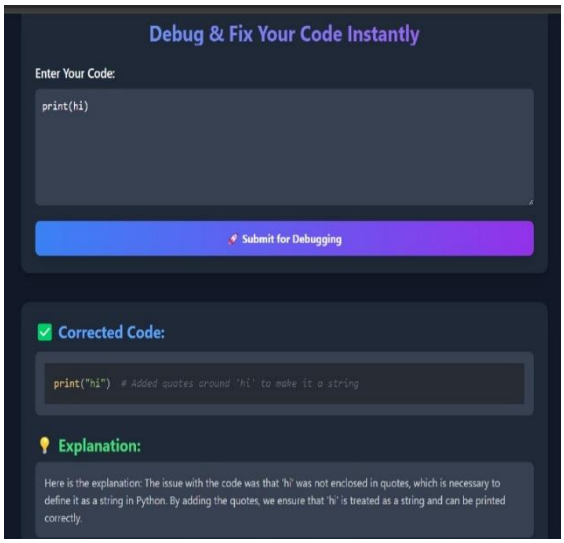

Fig. Debugging

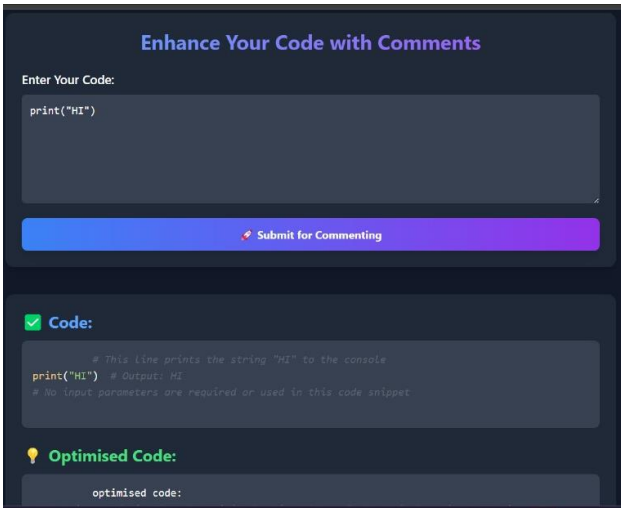

Fig. Comments

CHAPTER VII: REFERENCES

7.1 REFRENCES

The development of Code Genie was guided by various resources, including online documentation, AI frameworks, and developer tools. Below are some of the key references:

1. Programming Language Documentation

- JavaScript – MDN Web Docs (*https://developer.mozilla.org/en-US/docs/Web/JavaScript*)

2. AI API Integration Documentation

Groq API Documentation- https://console.groq.com/docs/

3. Software Development Best Practices

- Clean Code by Robert C. Martin (*Book on code readability and maintainability*)
- Stack Overflow, Medium, and Dev.to articles on best practices for code documentation and language translation

4. IDE & Code Editor References

- VS Code Extension Development – (*https://code.visualstudio.com/api*)