

Code Migrate AI

Ms Bhagyashree Basavaraj Kalavant¹, Ms Rashmi Amangonda Bagali², Dr Vanita Jadhav³,
Dr.Prajakta Satarkar⁴, Dr.Swati Pawar⁵

^{1,2,3,4,5}*Bachelor Of Technology, Department of Computer Science and Engineering,
SVRI's College of Engineering, Pandharpur*

Abstract—The growing diversity of programming languages has created significant challenges in software migration, code maintenance, and cross-platform development. Developers often spend considerable time manually rewriting code when transitioning between programming languages, increasing development costs and introducing potential errors.

The proposed platform integrates advanced AI models with modern web technologies to provide accurate, context-aware code translation [1].

The rapid growth of software development has increased the demand for efficient source code translation between different programming languages. Manual code conversion is often time-consuming and error-prone, requiring expertise in multiple programming languages. This paper presents an AI Multi-Language code Translator, a web-based application that automatically translates source code from one programming language to another using a Large Language Model (LLM). The proposed system integrates Node.js, Express.js, MongoDB, JWT Authentication, and Ollama with the Qwen 2.5 Coder model to provide fast, secure, and accurate code translation. The application Includes user registration and login with improved efficiency and scalability.

I. INTRODUCTION

In today's rapidly evolving software industry, multiple programming languages are used to develop applications across different platforms and domains. Each Programming language offers unique features, syntax, and capabilities that make it suitable for specific tasks. However, when developers need to migrate applications, maintain legacy systems, or collaborate on projects across different technologies, converting source code from one programming language to another can be challenging and time-consuming.[1]

Recent Advancements in Artificial Intelligence (AI) and Large Language Models (LLMs) have opened new possibilities for automated code generation and

translation. AI-powered systems can analyse programming structures, understand contextual relationships, and generate equivalent code in different programming languages with improved accuracy. These Capabilities reduce development time, improve productivity, and simplify the Software migration process.[2]

This project presents Code Migrate AI: An Intelligent Multi-Language Code Translation System, a web-based application designed to automatically convert source code between multiple programming languages, including C, C++. The system is developed using Node.js, Express.js, MongoDB, and HTML.[3] The proposed solution provides a simple and interactive user interface where users can enter source code, select a target programming language, and obtain translated code instantly.

Additionally, the system stores translation history in a database, enabling users to review and manage previous translations. By combining modern web technologies with Artificial Intelligence, the proposed system aims to provide an efficient, salable, and user-friendly solution for automated code translation.[4]

II. LITERATURE SURVEY

Automated code translation has become an important research area due to the increasing demand for software migration, maintenance of legacy systems, and cross-platform application development. Early code translation techniques primarily relied on rule-based systems that used predefined syntax mapping to convert source code from one programming language to another. Although these methods were effective for simple programs, they often struggled with complex programming structures and required significant manual effort to maintain and update translation rules. To overcome the limitations of rule-based approaches,

researchers introduced compiler-based translation techniques that utilize intermediate representations of source code.[1].

The emergence of Machine Learning and Artificial Intelligence has significantly transformed the field of code translation. Neural network-based models and sequence-to-sequence architectures enabled systems to learn programming patterns from large datasets and generate equivalent code in different languages. These approaches demonstrate better adaptability and contextual understanding compared to traditional methods. Recent advancements in Large Language Models (LLMs), such as GPT, DeepSeek, and similar AI models, have further improved code translation capabilities by understanding programming syntax, semantics, and logic at a deeper level [2].

The Large Language Models (LLMs) have significantly advanced automated code translation. These AI models can understand programming context and understand programming languages. They also support tasks such as debugging, code completion, and documentation generation. Their ability to learn from large code repositories makes them highly effective in software development.[3]

Despite these advancements, challenges remain due to differences in programming languages, syntax, memory management, and libraries. Translating complex concepts like

Multi-threading and language-specific APIs continues to require further research. Researchers are developing better optimization techniques to improve translation quality. The goal is to produce code that is efficient, maintainable, and semantically correct and optimize code more efficiently while reducing the time required for software development and maintenance. Researchers evaluate code translation systems using metrics such as syntax correctness, semantic equivalence, execution accuracy, and readability. These evaluation methods help improve the reliability and quality of AI-powered code translation tools. Human evaluation is also performed to ensure that the translated code follows programming standards and best practices. Continuous improvements in evaluation techniques help create more accurate and practical translation systems and user-friendly code translation for developers, students, and software professionals.[4]

III. PROPOSED APPROACH

There is a total of 6 modules, such as

- i) User Module
- ii) Source Code Input Module
- iii) Language Detection Module
- iv) AI Translation Engine
- v) Target Language Conversion Module
- vi) Admin Module

The proposed system, Code Migrate AI, is an Artificial Intelligence-based multi-language code translation platform designed to automate the conversion of source code between different programming languages. The system consists of several integrated modules that work together to provide accurate and efficient code translation. Initially, the user Module provides an interactive interface through which users can enter source code. The input Module validates the entered code and prepares it for processing. The Language Detection Module automatically identifies the source programming language by analyzing syntax patterns, keywords, and structural elements, thereby reducing manual effort and improving usability.[1]

The proposed system incorporates user authentication and secure data storage to protect user information and translation history. AI-powered processing minimizes manual coding effort and significantly reduces translation time. The modular architecture also allows for future enhancements, such as supporting additional programming languages and integrating newer AI models without major system modifications.[2]

The proposed system also maintains a translation history by automatically storing each successful translation in MongoDB Atlas. The stored information includes the source language, target language, source code, translation code, timestamp, and user details. This feature enables users to review previous translations, monitor their coding activities, and reuse translated programs without repeating the translation process. The cloud-based NoSQL architecture of MongoDB provides flexibility, scalability, and efficient management of translation records. Overall, the proposed AI Multi-Language Code Translator combines modern web technologies, cloud database management, secure authentication mechanisms.[3]

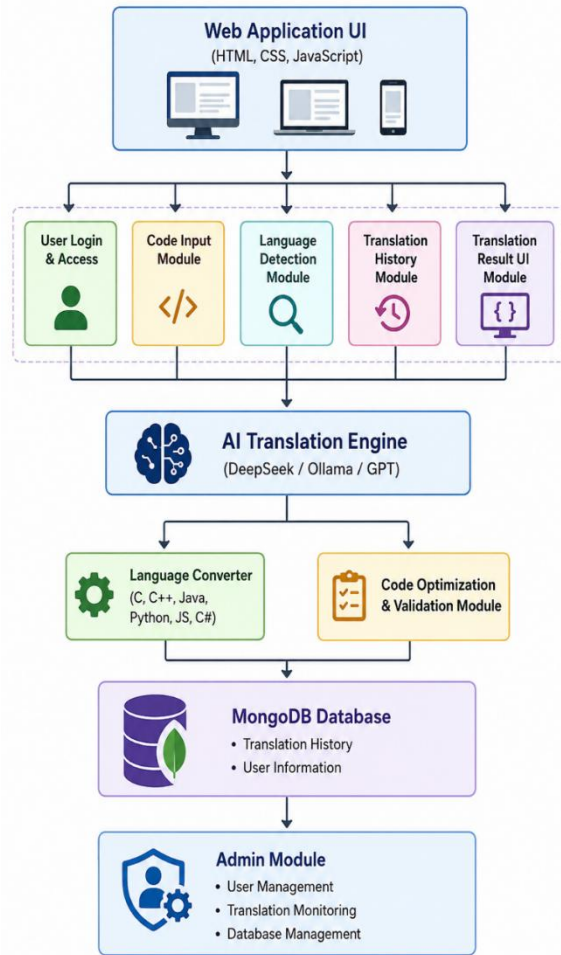


Figure 1- System Design

This diagram presents the architecture of an AI-powered Code Translation System designed to translate source code between multiple programming languages using advanced AI models such as DeepSeek, Ollama, or GPT. The system begins with a Web Application User Interface developed with HTML, CSS, and JavaScript, serving as the primary platform for user interaction. Through this interface, users can securely log in, enter or upload source code, explain the code, and access features such as automatic programming language detection, translation history, and translated code results.

Once the code is submitted, it is processed by the AI Translation engine, which analyses the source code and generates an equivalent version in the target programming language.

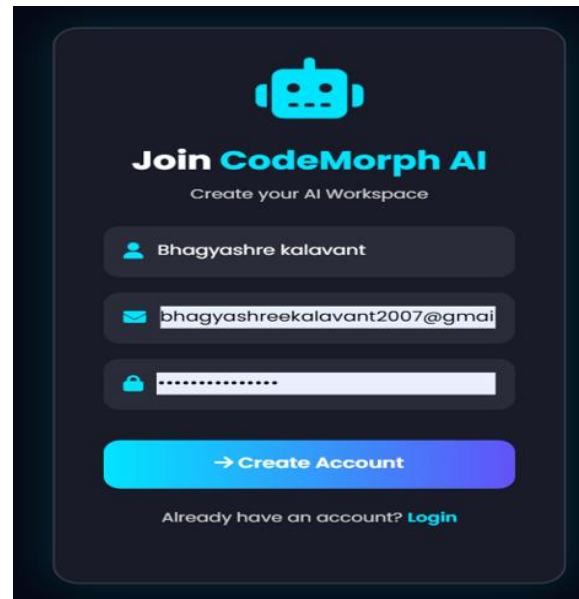


Figure 2- User Registers the Application

This Figure illustrates the user registration interface of the Code Morph AI web application, which serves as the entry point for new users to create an AI-powered coding workspace. The interface is designed with a modern and user-friendly layout that simplifies the account creation process. It includes input fields for the user's full name, email address, and password, enabling secure registration before accessing the system. A prominent "Create Account" button initiates the registration process, while the "Login" link allows existing users to access their accounts without creating a new one.

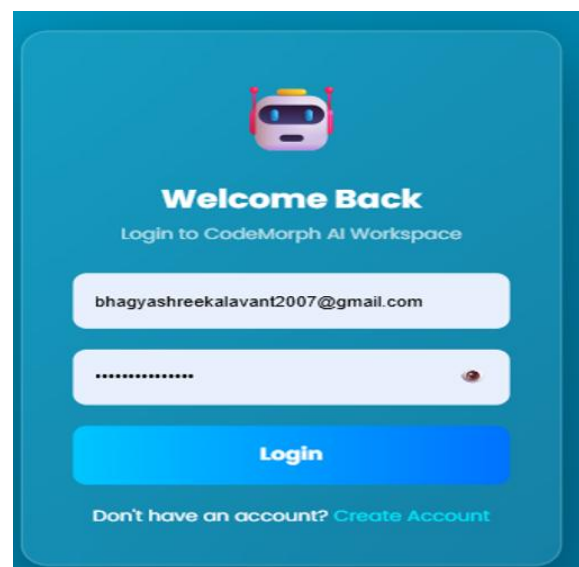


Figure 3- Use login the Application

This Figure illustrates the login interface of the Code Morph AI web application, which provides secure access to the AI-powered code translation workspace for registered users. The interface is designed with a clean and intuitive layout to ensure a seamless authentication experience. It consists of input fields for the users registered email address and password, along with a password visibility toggle that enhances usability while maintaining security.

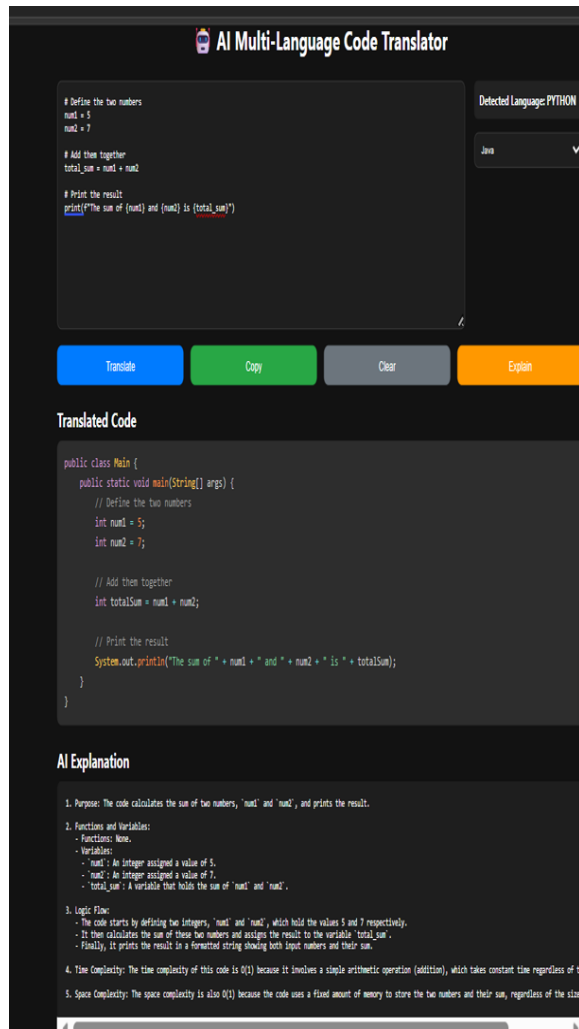


Figure 4-Display the translated Code to the Java
This image shows the user interface of an AI Multi-Language Code Translator web application designed with

A modern dark theme. At the top of the page, there is a title labelled “AI Multi Language Code Translator” along with an AI icon. The main section contains a large text editor where a Python program is entered.

The Python code defines two variables, num1=5 and num2=7, adds them together using total sum = num1 + num2, and prints the results using an f-string. On the right side of the editor, the application automatically detects the input language as python and provides a drop-down menu where the user has selected java as the targeted language for translation.

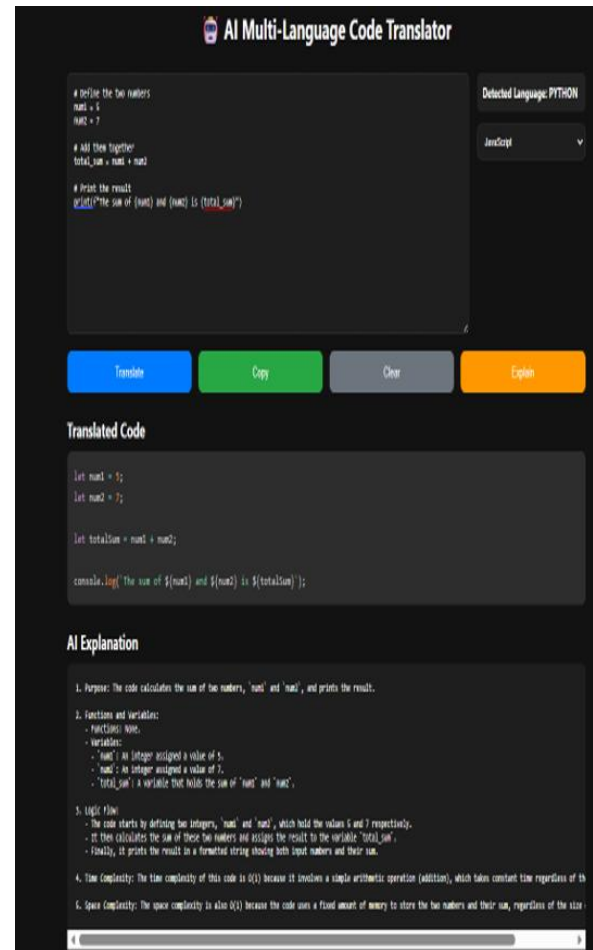


Figure .5-To display the translated Code to the JavaScript

This figure illustrates the graphical user interface of the AI Multi-Language Code Translator, a AI-powered web application developed to automatically translates source code between multiple programming languages while providing detailed explanations of the translated code. The interface is designed with a dark-themed layout to improve readability and user experience.

The Translated Code section presents the equivalent JavaScript implementation generated by the AI translation engine.

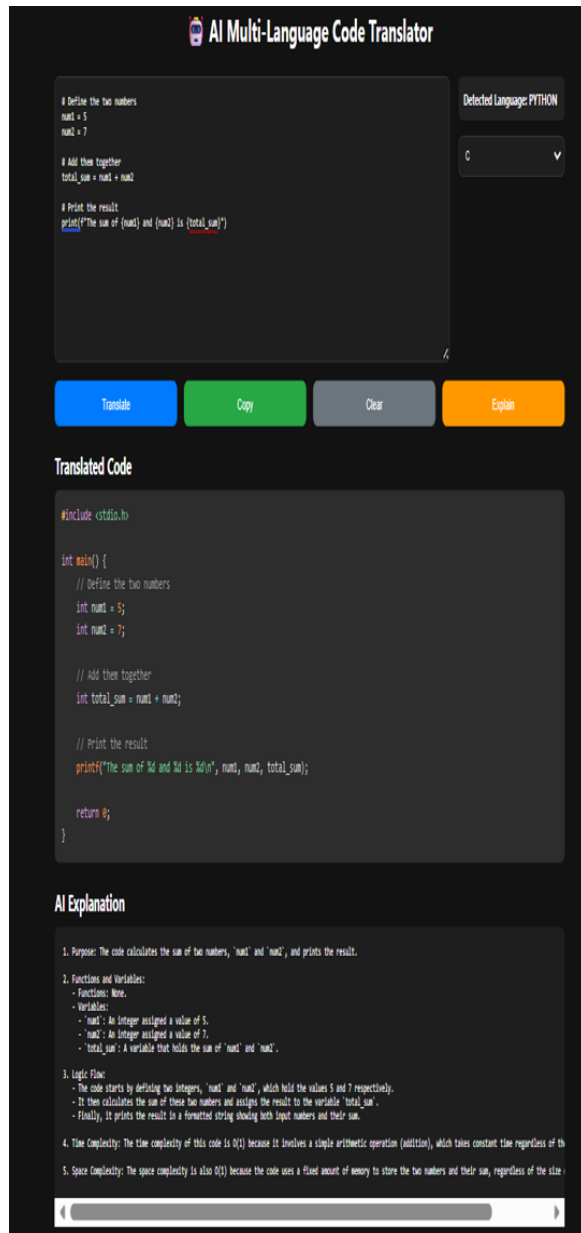


Figure .7 -To display the translated code to C

This figure illustrates the AI Multi-Language Translator translating a Python program into the C programming language. The application automatically detects Python as the source language and translates the input code while preserving its original logic and functionality. The translated output uses standard C syntax, including the `#include<stdio.h>` header file, the `main ()` function, integer variable declarations, arithmetic operations, and the `printf ()` function to display the result.

The interface also provides essential features such as Translate, Copy, Clear

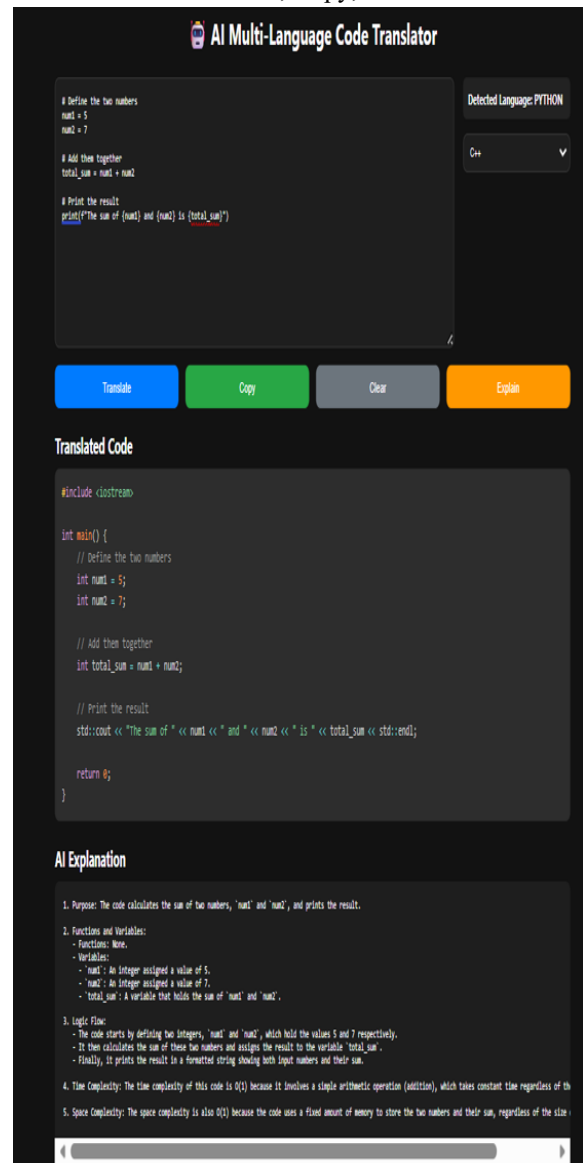


Figure 7- displays the translated code to C++

These Figure illustrates the AI Multi Code Transltor translatin a python program into the C++ programming language. The application automatically detects the source language as python and allows to the user to select C++ as the Targeted language through the language selection menu. The input program initializes two integer variables, calculates their sum, and displays the results using python's formatted string syntax.

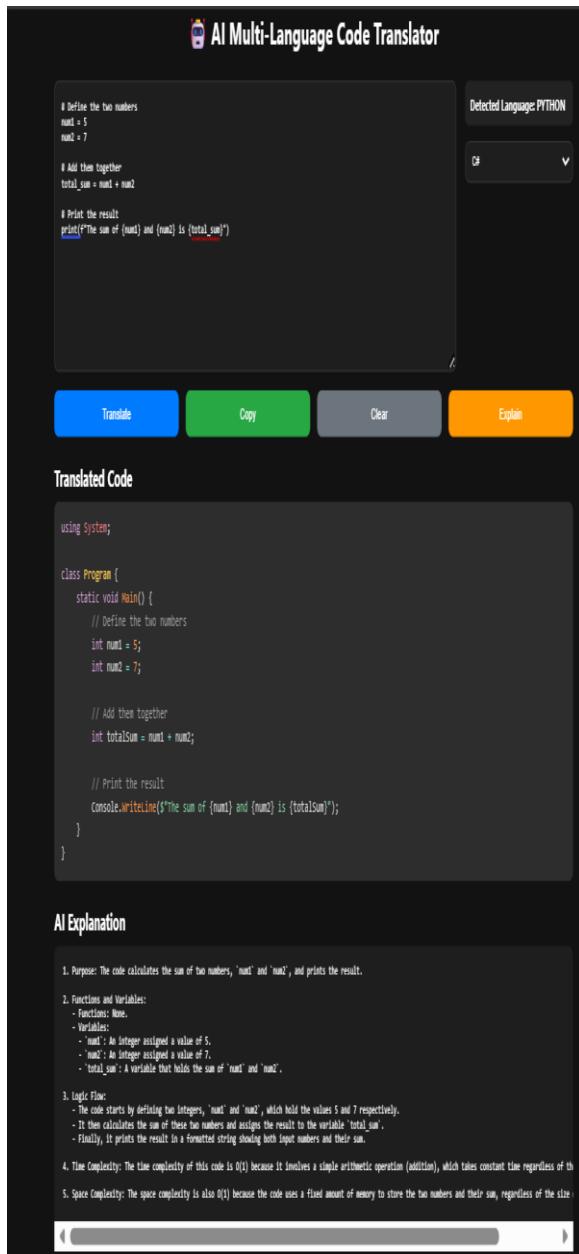


Figure 8- display the translated code to the C#

These Figure illustrates the AI Multi-Language code Translator translating a Python program into the C# programming language. The application automatically identifies Python as the source language and converts the code into equivalent C# syntax while preserving original logic and functionality. The translated program uses the System namespace, defines a Program class with the main() method, declares integer variables, performs the addition operation, and display the result using Console.WriteLine() with string interpolation.

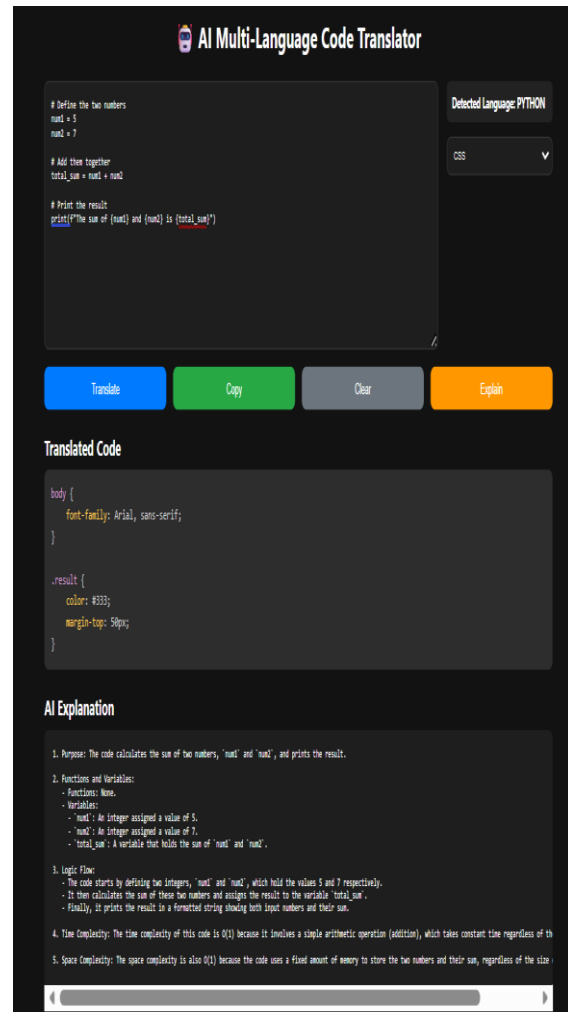


Figure .10- display the translated code to the CSS

This figure illustrates the AI Multi-Language Code Translator attempting to translate a Python program into CSS. The interface automatically detects Python as the source language and allows the user to select CSS as the target language from the drop-down menu. The input code is a simple Python program that defines two integer variables, calculates their sum, and prints the result using an f-string. The application provides a clean, dark-themed interface with syntax highlighting, making the source code easy to read and edit.

IV. CONCLUSION

In conclusion, the AI code Migrate AI is an intelligent web-based application designed to translate source code from one programming language to another using Artificial Intelligence. The system combines a user-

friendly interface with a powerful backend, enabling users to perform code translation.

The project demonstrates how AI models can simplify programming tasks by reducing manual code conversion and improving developer productivity. By interacting with technologies such as HTML, CSS, JavaScript, Node.js, Express.js, MongoDB, JWT Authentication, and Ollama AI.

The Ai Code Translator project demonstrates the practical application of Artificial Intelligence and modern web technologies in simplifying the process of programming language conversion. By integrating Large Language Models (LLMs) with a user-friendly web interface, the system enables developers, students, and researchers to translate source code accurately across multiple programming languages while preserving the original logic and functionality.

REFERENCES

- [1] OpenAI, "GPT-4 Technical Report," 2023.
- [2] DeepSeek AI, "DeepSeek Coder Technical Report," 2024.
- [3] Mozilla Developer Network, "JavaScript Guide." [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
- [4] T. Chen et al., "Evaluating Large Language Models for Code Generation," arXiv preprint, 2021.