PyScript: Python to JavaScript Converter

Sanika S. Bhondivale[1], Yogita N. Kadav[2], Sahil S. Bhondiwale[3], Varun R. Jadhav[4]
[1,2,3,4] *Alamuri Ratnamala Institute of Engineering and Technology*

***Abstract*—PyScript is a powerful web application developed for both beginner as well as expert developers that consistently translates Python code into JavaScript. Basic Python scripts, complex multi-threading, object-oriented programming using classes, and even external library integration are just a few of the many features it manages. For projects of different levels of complexity, this makes it the perfect tool. The Convert & Run option is one of its primary advantages; it allows users to run the converted JavaScript code immediately and see the results in real time. PyScript also comes with a powerful debug tool that makes debugging much simpler by identifying issues, analysing difficult code lines, and suggesting solutions. The application offers real-time feedback on outputs and mistakes for ensuring that the process runs smoothly. Additionally, PyScript improves the coding experience by offering options to download or share the converted JavaScript code for future use or collaboration, as well as syntax highlighting for improved readability. Its clean animations and recent, attractive style combine to create an experience that is simple to use and accessible. PyScript is an essential tool for developers working with Python and JavaScript since it simplifies the process with high accuracy and advanced functionality, regardless of the complexity of the application or the script being converted.**

***Index Terms*—Python to JavaScript, Real-time feedback, User-friendly, Efficiency, Download and Share code.**

I. INTRODUCTION

An innovative web-based application called PyScript is created to help developers convert code between Python and JavaScript in a smooth and efficient manner. PyScript, which is design with efficiency and accuracy in mind, is a dependable tool for developers working on a variety of coding tasks since it can translate Python code to JavaScript with over 90% correctness. PyScript manages multi-threading, object-oriented programming, external library integration, and other complex tasks with simplicity, regardless of how basic your scripts are. Its user-friendly interface, which includes a Convert & Run option to run converted code immediately. The application is appropriate for both beginners and experts because to its real-time feedback. PyScript makes it easier to collaborate and manage workflows by providing easy ways to download or distribute converted code.

PyScript is a flexible tool that speeds up development and makes Python-to-JavaScript conversion accessible to all users by combining innovative capability with a contemporary, user-friendly interface. Along with its primary functions, PyScript also enables batch conversions, which saves developers a significant amount of time on bigger projects by enabling them to convert several Python scripts to JavaScript simultaneously. Whether working with simple syntax or more complex structures, users can modify the translation process according to their particular requirements due to the application's customizable options. Even complicated Python functions, classes, and modules are reliably transformed while retaining their intended functionality because of PyScript's creative code analysis. The program also integrates version control, enabling developers to monitor modifications and roll back to previous versions as necessary. PyScript is designed to keep up with the most recent developments in Python and JavaScript programming through regular updates and improvements.

PyScript supports the conversion of Python code to JavaScript with easy-to-understand. It handles complicated code with ease and is suitable for both small and big projects. Developer productivity is improved by the real-time feedback, which allows them to observe changes immediately. PyScript's user-friendly architecture makes it perfect for both beginners and experts, saving time and reducing frustration. PyScript is a useful tool for developers that want consistent quick results since it simplifies the conversion process.

II. PROBLEM STATEMENT

For developers working on cross-platform applications or projects that need both languages, the challenge and time-consuming nature of translating Python code into JavaScript is the issue this project attempts to solve. Even though Python and JavaScript have separate syntaxes and different uses, many developers find it difficult to switch between them. When working on big projects, this can be especially difficult because it often requires manually rewriting huge sections of code, which can result in mistakes, inconsistencies, and extra development time.

Additionally, developers have to handle complex characteristics that make translation more difficult, such as object-oriented programming, multi-threading, and external library integration. Additionally, there aren't many effective tools available for batch processing of several scripts, or real-time feedback. The requirement for an accessible solution that allows for both beginners and experienced developers causes the issue. While experienced developers may find it difficult to manage large-scale, complex code conversions efficiently, new developers may find it difficult to understand and resolve conversion issues.

Thus, the objective of this project is to produce a user-friendly web-based tool called PyScript that enables developers to easily handle multi-threading and object-oriented features, give real-time feedback, and transform Python code into JavaScript with high accuracy. This tool is designed to reduce the frustration of converting between programming languages, increase productivity, and simplify the development method, particularly for larger projects that call for complex functions or batch conversions.

III. LITERATURE SURVEY

Matsiievskyi, Oleksii, et al. (2024): - Proposed AI-based methods for converting code between different programming languages, still under development; may not support all languages and contexts.[1]

Lopes, Mateus, and Andre Hora et al. (2022): - Studied the causes of complex methods in software development across multiple languages, Focuses on empirical analysis, not on algorithmic solutions.[2]

Ahmad, Wasi Uddin, Md Golam Rahman Tushar, Saikat Chakraborty, Kai-Wei Chang et al. (2021): - Created a parallel corpus for Java-Python program translation using machine learning, Limited to the scope of Java and Python; requires refinement for broader languages.[3]

Gagniuc, Paul A. et al. (2020): - Provided an introductory framework for learning multiple programming languages simultaneously, Focuses on learning methods rather than specific algorithms.[4]

Jin, Eric and Yu Sun et al. (2020): - Developed an algorithm-adaptive source code converter for translating Python to Java, May not handle all edge cases in translation; focused on Python to Java.[5]

IV. MAIN PURPOSE

This project's primary goal is to create PyScript, a web tool that lets programmers easily and accurately translate Python code into JavaScript. By offering smooth translation, real-time feedback, and support for advanced technologies like multi-threading, object-oriented programming, and external library integration, the tool aims to make the coding process easier. PyScript helps developers of all skill levels save time, increase productivity, and better manage cross-platform projects by providing an intuitive interface with customisable factors and batch conversion features.

PyScript also makes it easier for developers to share and download code that has been transformed, which improves teamwork. Users can keep track of changes, roll back to earlier iterations when necessary, and preserve an organized development process with integrated version control. For developers working on a variety of projects, PyScript remains a dependable and current tool thanks to regular upgrades that maintain it in line with the most recent developments in Python and JavaScript.

The organized workflow of PyScript reduces the frustration of manual code conversion, allowing programmers to focus more on logic and functionality rather than syntax differences. PyScript, which makes the transition between Python and JavaScript easier, enables developers to build web applications, automate tasks, and experiment with new ideas more efficiently. PyScript is made to be both powerful and user-friendly, making it accessible

to beginners while offering advanced features for experienced developers.

V. OBJECTIVE

The PyScript project's goal is to provide a web-based tool that allows Python code to be converted into JavaScript with ease and accuracy—more than 90% correctness. The tool is made to give developers real-time feedback during the conversion process, enabling them to make changes right away. It aims to manage difficult programming ideas while maintaining code integrity, including multi-threading, object-oriented programming, and external libraries. Additionally, PyScript allows for batch conversion, which greatly increases productivity, particularly for larger projects, by allowing developers to convert multiple scripts at once.

The program has a user-friendly interface that is appropriate for beginners as well as professionals, and it provides customisable conversion choices to accommodate different coding requirements. It integrates version control for tracking changes and allows users to revert to previous versions if necessary. PyScript is designed to stay up-to-date with the latest programming advancements, streamlining the development process, reducing manual effort, and ultimately enhancing the overall efficiency of developers.

Another fundamental purpose of PyScript is to bridge the gap between Python and JavaScript programming, making cross-platform writing more accessible and efficient. By automating the conversion process, PyScript reduces the need for developers to manually rewrite code, reducing errors and saving critical time. It gives programmers the ability to easily incorporate Python logic into JavaScript-based projects, making tasks like data visualization, web development, and automation easier. PyScript strives to be a dependable and future-ready solution for developers working in a variety of programming contexts through its constant enhancements and flexibility to changing coding standards.

VI. SCOPE OF PROJECT

The goal of the PyScript project is to provide a complete web-based platform for the easy translation of Python to JavaScript code. The ability to run the converted code using a "Convert & Run" tool, accurate conversion with over 90% success rate, and strong debugging help that identifies and explains issues with exact line numbers and thorough descriptions are some of its primary features. It also improves code readability by highlighting syntax for both Python and JavaScript, and it gives real-time feedback on output or problems.

From simple Python scripts to more complicated situations like classes, object-oriented programming, multi-threading, and interaction with external libraries, the platform is capable of handling it all. PyScript incorporates capabilities like download and share options for smooth teamwork to improve user interaction.

The project provides for a broad range of users, including experts who need complex code translation help to increase productivity and novice developers looking for an easy-to-use interface. The design demonstrates on usability, having an innovative, responsive, and user-friendly interface with syntax-aware colour coding, subtle animations, and a clean layout.

PyScript efficiently manages runtime and logic-based errors, supports the most recent versions of Python and JavaScript, and permits frequent updates to preserve accuracy and incorporate new features. The project's main goals are to create an interactive learning tool for Python and JavaScript differences, optimize developers' workflows, and lessen the effort needed for manual code conversion. PyScript makes a name for itself as a flexible and inclusive tool that caters to a wide range of coding requirements by handling both beginner and expert use cases.

VII. METHODOLOGY

To provide an intuitive user interface that can handle complex Python-to-JavaScript conversions, the PyScript project takes a rigorous approach that starts with demand research and user interface design. The primary aim is to develop a translation engine that parses Python code and converts it into JavaScript while handling issues like multi-threading and external libraries. Real-time error detection during conversion and execution, providing users with comprehensive feedback. The converted code may run right in the browser thanks to the "Convert &

Run" feature. The UI includes syntax highlighting and interactive elements for a smooth experience. Extensive testing after development ensures correctness and functionality. After then, the product is released and enhanced in response to consumer feedback. Long-term effectiveness will be ensured by routine maintenance and upgrades.

VIII. PROPOSED SYSTEM

The user starts the process by entering Python code into an interface text box or input field. The conversion procedure is started when the user clicks the "Convert & Run" button after the code is ready. This action makes a request to the backend, which controls the conversion process as a whole. The Python code is sent via the conversion engine by the backend to verify its accuracy before processing the request. It successfully converts the code into JavaScript and returns it to the frontend for display if there are no errors. But if there are any problems, the code is sent to the debugging engine, which inspects the mistakes and produces the relevant error messages. The output is then shown by the frontend, which helps the user understand and efficiently fix any possible errors by displaying either the converted JavaScript code if everything is correct or an error message if any difficulties were found.

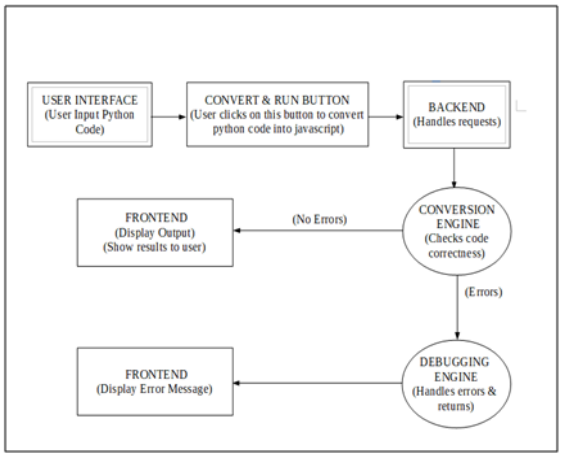

Fig 1. Proposed System

IX. SYSTEM ARCHITECTURE

A web-based tool that automatically converts Python code to JavaScript is the suggested system for the PyScript project. With support for both simple structures and more sophisticated features like classes, multi-threading, and external libraries, it has a Python-to-JavaScript conversion engine that reliably converts Python syntax. Issues in the converted code will be highlighted by real-time error detection. Users will be able to convert Python code into JavaScript and run it straight in the browser with instant feedback thanks to the "Convert & Run" functionality. The user interface will be dynamic and contemporary, with color coding and syntax highlighting for improved reading. Additionally, users will be able to download or share their transformed code. The switch from Python to JavaScript for web applications is made easier using this solution.

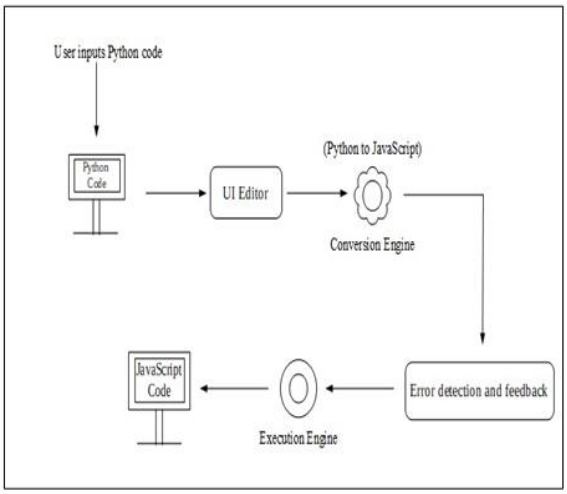

Fig 2. System Architecture of PyScript

X. DETAILS OF DESIGN, WORKING AND PROCESS

STEP 1: Enter Python code into PyScript.
Write or paste your Python code into the PyScript interface to get started. You can enter the script you wish to turn into a local application or a web-based tool to achieve this.

STEP 2: PyScript automatically converts it to JavaScript.
PyScript interprets the Python code once you enter it and converts it to JavaScript syntax. Since this stage is immediate seeing the converted version is simple and requires no additional work.

STEP 3: Receive real-time feedback.

The tool instantly displays the equivalent JavaScript code after the conversion. This helps users in understanding the JavaScript translation of their Python script.

STEP 4: Convert multiple scripts at once.
The tool instantly displays the equivalent JavaScript code after the conversion. This helps users in understanding the JavaScript translation of their Python script.

STEP 5: Customize the conversion settings.
Certain options can be changed by users, such as selecting alternative JavaScript frameworks (such Vanilla JS or Node.js) or modifying the code to certain use cases. The above settings help in modifying the output according to project requirements.

STEP 6: Download or share the JavaScript code.
After the conversion is finished, users can share the JavaScript file directly with others using an integrated sharing tool, or they can download it for later use. Importing the translated code into other projects is made simple as a result.

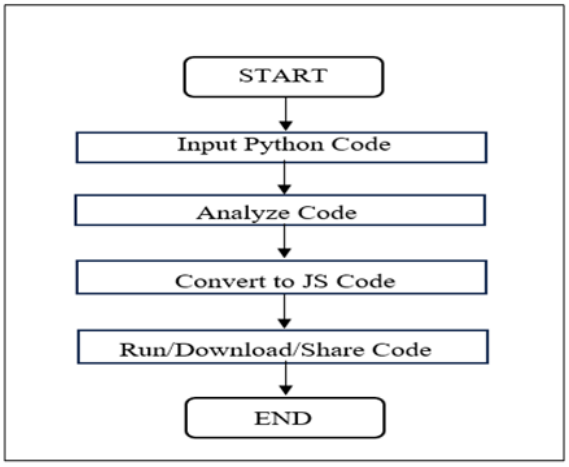

Fig 3. Flowchart of PyScript

XI. RESULTS AND CONCLUSION

A Python-to-JavaScript converter that converts Python code to JavaScript is shown in the image. A loop through a list, a sum operation, and a print statement are all included in the Python code on the left. While the sum procedure adds two numbers and uses `format () ` to produce the result, the print statement shows `"Hello, world!"`. The loop uses an f-string to output each value as it iterates through a list of numbers. The converted JavaScript code on the right has a similar structure, with the loop modified for JavaScript syntax and `console.log () ` in place of `print () `. Python scripts can be easily converted to JavaScript for web-based applications with this utility.

Fig 4. Output of PyScript

This table examines whether Python features such as print statements, loops, functions, and error handling are accurately translated when a Python-to-JavaScript converter is used. A fragment of Python code, its anticipated JavaScript equivalent, and the success or failure of the conversion are displayed in each row. Every test case passed, indicating that the converter functions well in these fundamental situations.

Test Case	Input Python Code	Expected Output	Result
Print Statement	print("Hello")	console.log("Hello");	Pass
Input to Prompt	name = input("Enter name: ")	let name = prompt("Enter name: ");	Pass
Simple Function	def greet(name): return "Hello " + name	function greet(name) { return "Hello " + name; }	Pass
For Loop	for i in range(5): print(i)	for (let i = 0; i < 5; i++) { console.log(i);	Pass

		}	
If-Else Statement	if x > 10: print("Greater")	if (x > 10) { console.log("Greater"); }	Pass
List Comprehension	[x**2 for x in range(5)]	rray.from({length: 5}).map((x) => x**2);	Pass
Try-Except	try: 1/0 except: print("Error")	try { 1/0; } catch (error) { console.log("Error"); }	Pass
String Reverse	"hello"[::-1]	"hello".split('').reverse().join('')	Pass

Table 1. Test Case Table

PyScript is an innovative web-based tool that converts Python to JavaScript code quickly and accurately—more than 90%. It is efficient and user-friendly, with support for external libraries, object-oriented programming, and multi-threading. It also offers debugging tools, syntax highlighting, and real-time feedback. Productivity is increased by its batch conversion capability, and workflow management is made easy by its inbuilt version control. Python-to-JavaScript translation is smooth, effective, and accessible for a variety of coding tasks thanks to PyScript's constant evolution with upgrades, which makes it a dependable option for both beginners and experts.

REFERENCES

[1] Matsiievskyi, Oleksii, et al. (2024): - Proposed AI-based methods for converting code between different programming languages, still under development; may not support all languages and contexts.

[2] Lopes, Mateus, and Andre Hora et al. (2022): - Studied the causes of complex methods in software development across multiple languages, Focuses on empirical analysis, not on algorithmic solutions.

[3] Ahmad, Wasi Uddin, Md Golam Rahman Tushar, Saikat Chakraborty, Kai-Wei Chang et al. (2021): - Created a parallel corpus for Java-Python program translation using machine learning, Limited to the scope of Java and Python; requires refinement for broader languages.

[4] Gagniuc, Paul A. et al. (2020): - Provided an introductory framework for learning multiple programming languages simultaneously, Focuses on learning methods rather than specific algorithms.

[5] Jin, Eric and Yu Sun et al. (2020): - Developed an algorithm-adaptive source code converter for translating Python to Java, May not handle all edge cases in translation; focused on Python to Java.