Project Implementation and Testing

Date	8th July 2024
Team ID	SWTID1720074725
Project Name	CodeXchange: An AI-Powered Code Translator Tool

Testing and Evaluation

Testing Strategy

The tool will be rigorously tested using a diverse set of code snippets in Python, Java, and C++ to ensure both accuracy and functionality. Testing will encompass both the user interface (Streamlit) and the command-line interface (CLI), focusing on performance, reliability, and user experience.

Test Cases and Results

Test cases will cover a wide range of scenarios, including:

1. **Simple Functions**: Basic code snippets such as "Hello, World!" programs and simple arithmetic operations.
2. **Complex Algorithms**: More intricate code involving data structures, algorithms, and libraries.
3. **Edge Cases**: Scenarios with syntax errors, incomplete code snippets, and unconventional coding styles.

Each test case will be evaluated for:

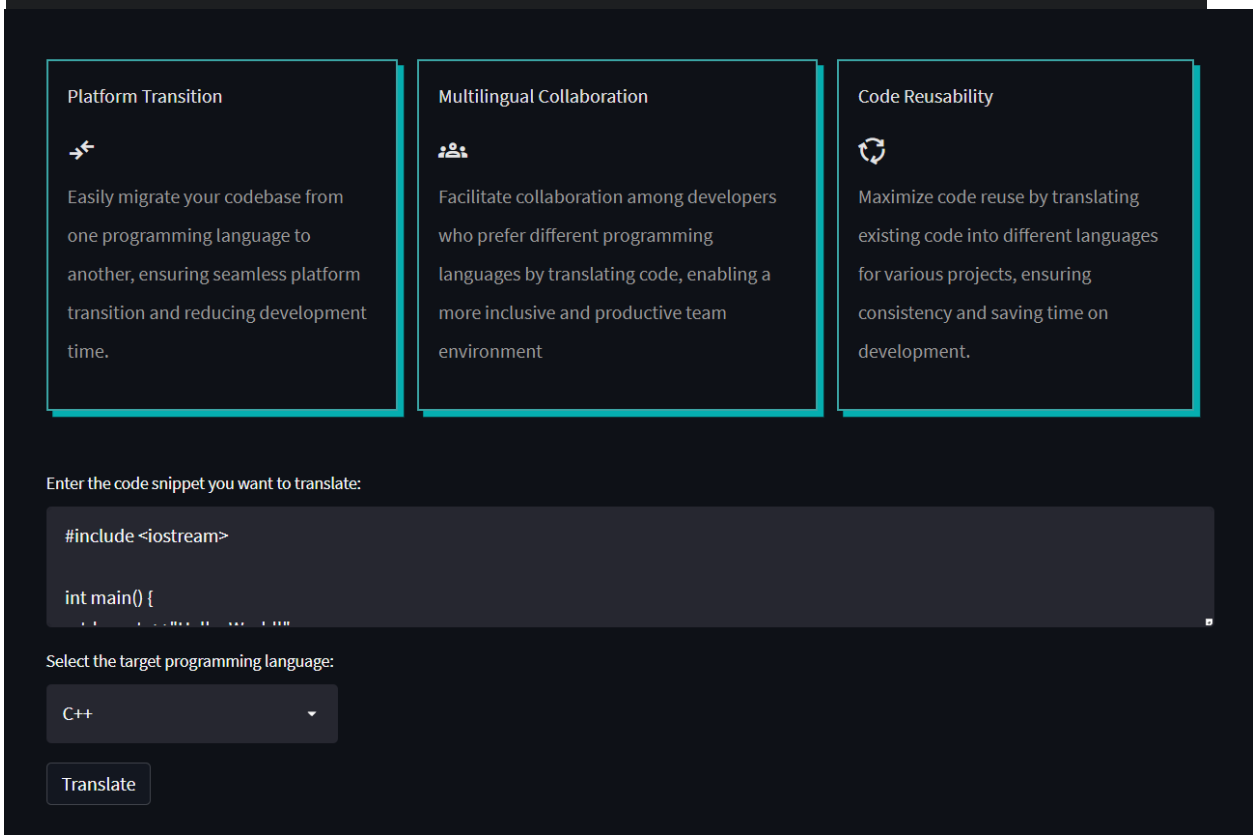
- **Translation Accuracy**: Ensuring the translated code maintains the same logic and functionality as the original.
- **Error Handling**: Verifying that the tool appropriately handles syntax errors and incomplete code.
- **Performance**: Assessing the speed and responsiveness of the translation process.

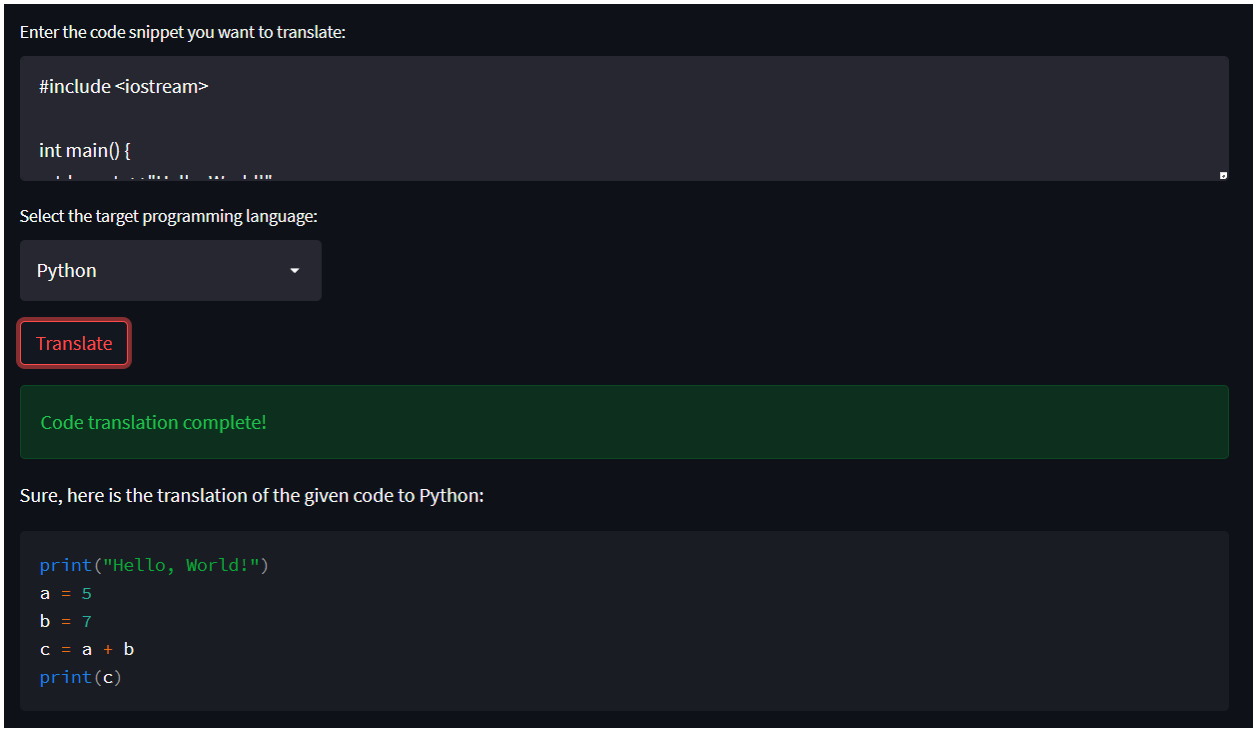
Results will be analyzed to ensure high accuracy and reliability. Any identified issues will be addressed through iterative adjustments and refinements to the tool.

Sample Test Cases:-

Test Case 1:-

```
1  #include <iostream>
2
3  int main() {
4    std::cout << "Hello, World!";
5    int a = 5;
6    int b = 7;
7    int c = a + b;
8    std::cout << c;
9    return 0;
10 }
```

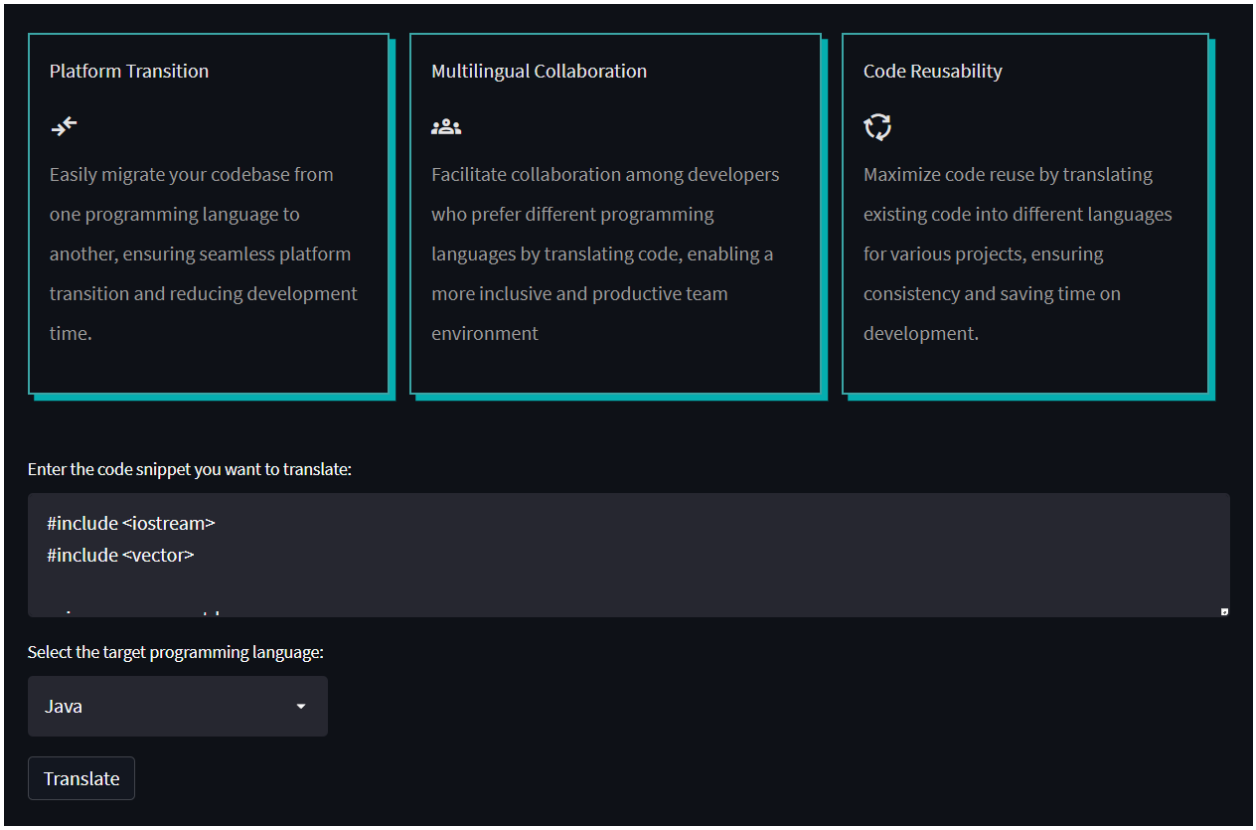


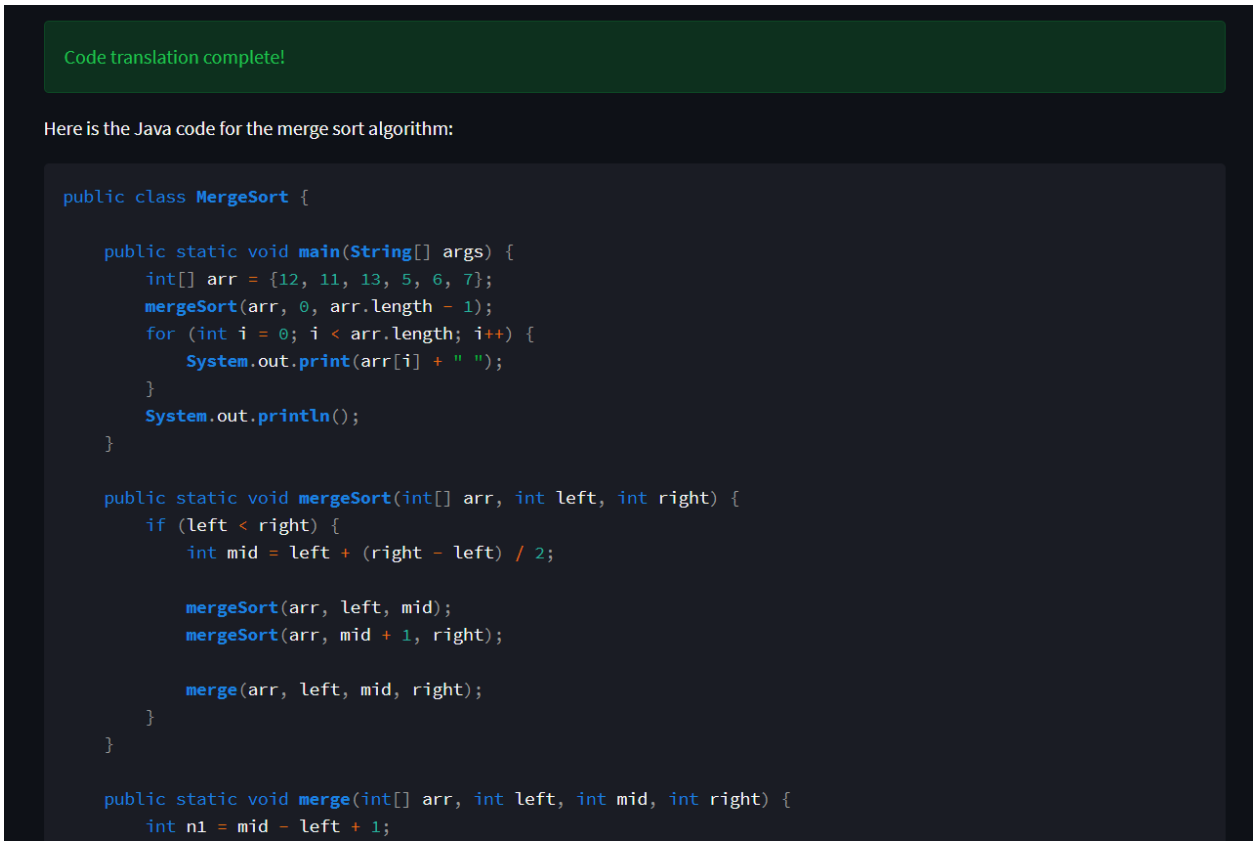


Test Case 2:-

```
// Merge Sort Algorithm
#include <iostream>
#include <vector>

using namespace std;

void merge(vector<int>& arr, int left, int mid, int right) {
    int n1 = mid - left + 1;
    int n2 = right - mid;

    vector<int> leftArray(n1);
    vector<int> rightArray(n2);

    for (int i = 0; i < n1; ++i)
        leftArray[i] = arr[left + i];
    for (int j = 0; j < n2; ++j)
        rightArray[j] = arr[mid + 1 + j];

    int i = 0, j = 0, k = left;
    while (i < n1 && j < n2) {
        if (leftArray[i] <= rightArray[j]) {
```

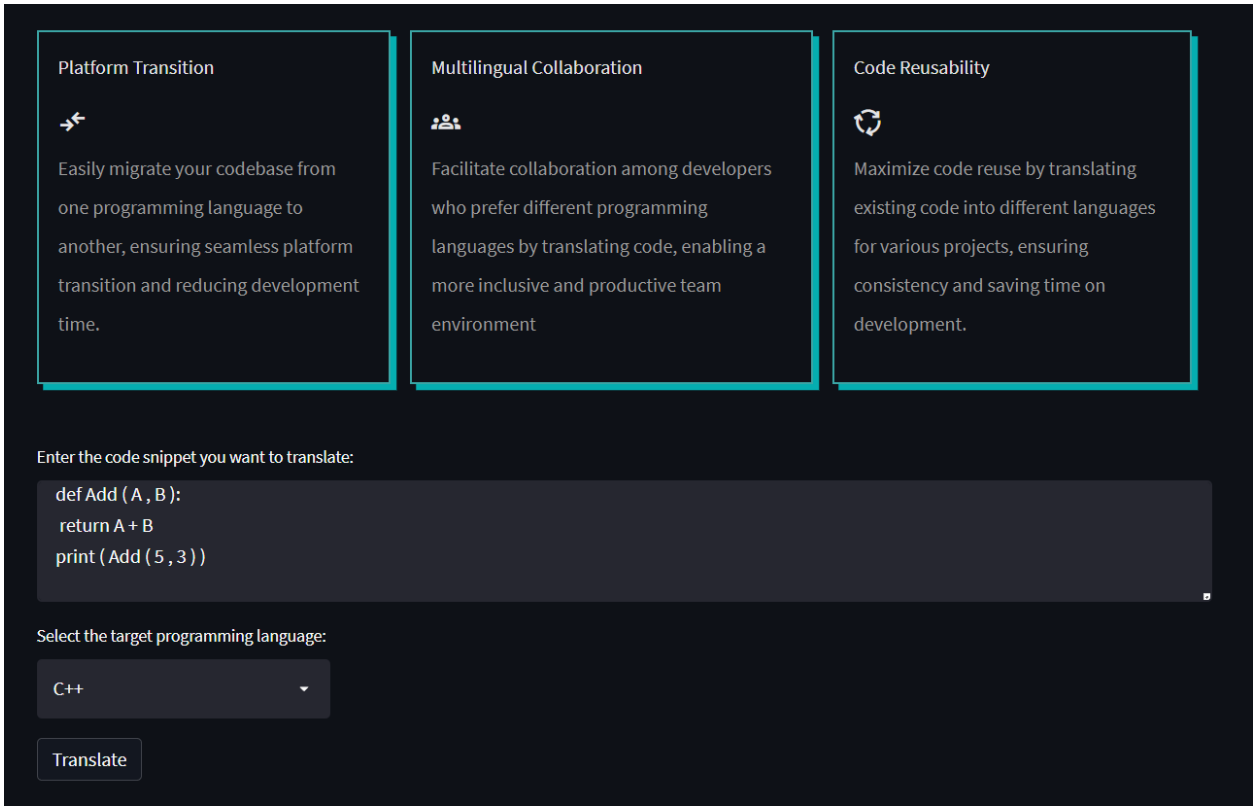
```
            arr[k] = leftArray[i];
            i++;
        } else {
            arr[k] = rightArray[j];
            j++;
        }
        k++;
    }

    while (i < n1) {
        arr[k] = leftArray[i];
        i++;
        k++;
    }

    while (j < n2) {
        arr[k] = rightArray[j];
        j++;
        k++;
    }
}

void mergeSort(vector<int>& arr, int left, int right) {
    if (left < right) {
        int mid = left + (right - left) / 2;

        mergeSort(arr, left, mid);
        mergeSort(arr, mid + 1, right);

        merge(arr, left, mid, right);
    }
}

int main() {
    vector<int> arr = {12, 11, 13, 5, 6, 7};
    mergeSort(arr, 0, arr.size() - 1);
    for (int i = 0; i < arr.size(); i++)
        cout << arr[i] << " ";
    cout << endl;
    return 0;
}
```


Platform Transition
Easily migrate your codebase from one programming language to another, ensuring seamless platform transition and reducing development time.
Multilingual Collaboration
Facilitate collaboration among developers who prefer different programming languages by translating code, enabling a more inclusive and productive team environment
Code Reusability
Maximize code reuse by translating existing code into different languages for various projects, ensuring consistency and saving time on development.
Enter the code snippet you want to translate:
#include <iostream>
#include <vector>
Select the target programming language:
Java
Translate


Code translation complete!
Here is the Java code for the merge sort algorithm:
public class MergeSort {
public static void main(String[] args) {
int[] arr = {12, 11, 13, 5, 6, 7};
mergeSort(arr, 0, arr.length - 1);
for (int i = 0; i < arr.length; i++) {
System.out.print(arr[i] + " ");
}
System.out.println();
}
public static void mergeSort(int[] arr, int left, int right) {
if (left < right) {
int mid = left + (right - left) / 2;
mergeSort(arr, left, mid);
mergeSort(arr, mid + 1, right);
merge(arr, left, mid, right);
}
}
public static void merge(int[] arr, int left, int mid, int right) {
int n1 = mid - left + 1;

Test Case 3:-

```
# Syntax Error
def add(a, b):
    return a + b

print(add(5, 3)

# Incomplete Code Snippet
def multiply(a, b):
    return

# Unconventional Coding Style
def Add ( A , B ):
 return A + B
print ( Add ( 5 , 3 ) )
```



Select the target programming language:

C++

Translate

Code translation complete!

Sure, here is the translated code:

```cpp
#include <iostream>

using namespace std;

int add(int a, int b) {
    return a + b;
}

int main() {
    cout << add(5, 3) << endl;
    return 0;
}
```