INDUSTRIAL INTERNSHIP PROJECT DOCUMENTATION

CodeXchange: An AI-Powered Code Translator Tool Using Palm's Chat-Baison-001

Course : **Generative AI Applications using Vertex AI in collaboration with Google**

Team Id : **SWTID1720596651**

Team Members :

K. Madhulika

M.Sharath Chandra

T.Vishnu Gopal Reddy

Aditya.S

Project Description

CodeXchange is an innovative web application designed to streamline code translation and facilitate seamless collaboration among developers working with different programming languages. Whether you're transitioning applications between platforms, collaborating in multilingual teams, or reusing code across projects, CodeXchange empowers developers to effortlessly translate code snippets between various programming languages. Leveraging advanced translation algorithms and syntax analysis, CodeXchange ensures accurate and reliable code conversion while preserving the original functionality and logic. With its intuitive interface and comprehensive language support, CodeXchange revolutionizes the development workflow, enabling teams to work together efficiently, enhance code reusability, and accelerate project delivery.

Scenario 1: Platform Transition

CodeXchange assists developers in transitioning applications from one platform to another. For instance, a team working on an application written in Python needs to migrate it to Java to leverage Java's robustness and scalability in an enterprise environment. By inputting the Python code snippets and selecting Java as the target language, developers receive accurately translated code that maintains the original functionality, streamlining the migration process and minimizing the risk of introducing errors.

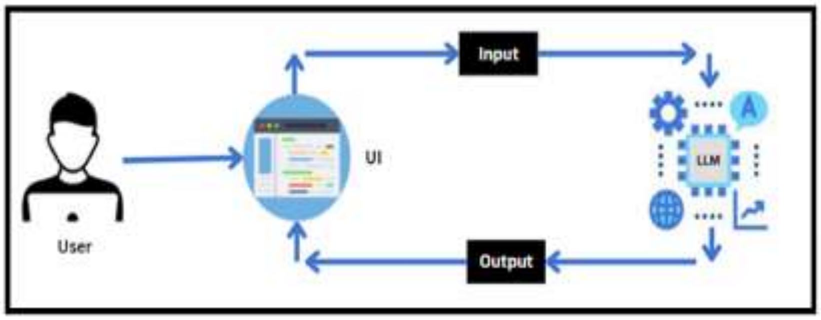
Scenario 2: Multilingual Collaboration

In a collaborative project where team members use different programming languages, CodeXchange facilitates seamless integration by translating code snippets as needed. Suppose one part of the team is proficient in C++ while another prefers Python. Developers can write code in their preferred language and use CodeXchange to translate it, ensuring all team members can work together efficiently without being constrained by language barriers. This enhances productivity and reduces the learning curve associated with adopting new languages.

Scenario 3: Code Reusability Across Projects

CodeXchange promotes code reusability by enabling developers to translate existing code into different languages for new projects. For example, a developer has written a set of utility functions in Java that would be beneficial for a new project being developed in C++. By translating these Java functions into C++ using CodeXchange, the developer can quickly integrate proven code into the new project, saving time and ensuring consistency across different projects.

Technical Architecture



Project Flow

- Users input text into the UI of Inquisitive.
- The input text is then processed and analyzed by the PALM architecture, which is integrated into the backend.

- PALM autonomously generates questions based on the input text.
- The generated questions are sent back to the frontend for display on the UI.
- Users can view the dynamically generated questions and interact with them to gain deeper insights into the content.

To accomplish this, we have to complete all the activities listed below,

- Initializing the PALM
    - Generate PALM API
    - Initialize the pre-trained model
- Interfacing with Pre-trained Model
    - Questions Generator
- Model Deployment
    - Deploy the application using Streamlit

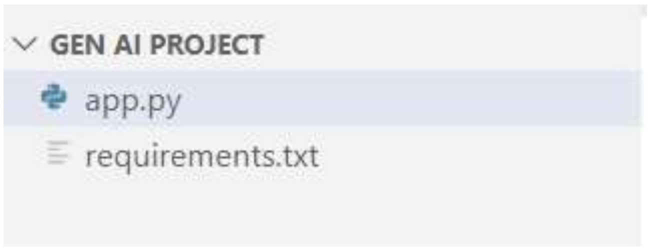
Prior Knowledge

You must have prior knowledge of the following topics to complete this project.

- LLM & PALM: https://cloud.google.com/vertex-ai/docs/generative-ai/learn-resources
- Streamlit: https://www.datacamp.com/tutorial/streamlit

Project Structure

Create the Project folder which contains files as shown below:



```
GEN AI PROJECT
  app.py
  requirements.txt
```

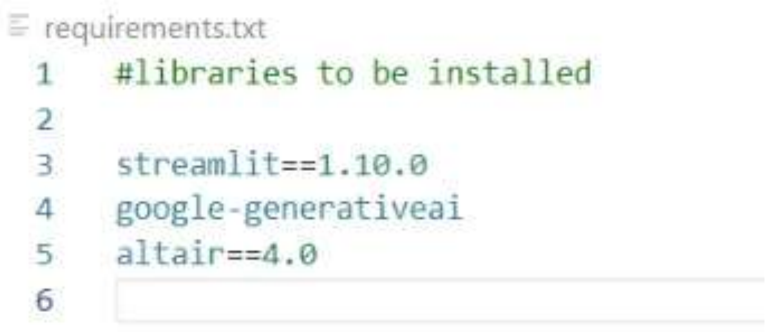
- app.py: It serves as the primary application file housing both the model and Streamlit UI code.
- requirements.txt: It enumerates the libraries necessary for installation to ensure proper functioning.

MILESTONE 1 : REQUIREMENT SPECIFICATION

Specifying the required libraries in the requirements.txt file ensures seamless setup and reproducibility of the project environment, making it easier for others to replicate the development environment.

ACTIVITY 1

Create A Requirements.Txt File To List The Required Libraries.



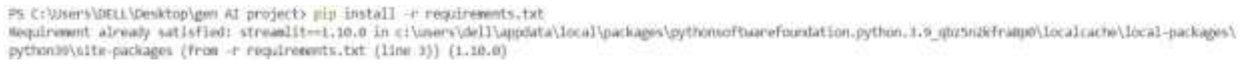
requirements.txt

```
#libraries to be installed

streamlit==1.10.0
google-generativeai
altair==4.0

```

- streamlit: Streamlit is a powerful framework for building interactive web applications with Python.
- google-generativeai: Python client library for accessing the GenerativeAI API, facilitating interactions with pre-trained language models like Gemini Pro.

ACTIVITY 2

Install The Required Libraries



```
PS C:\Users\DELL\Desktop\gen AI project> pip install -r requirements.txt
Requirement already satisfied: streamlit==1.10.0 in c:\users\dell\appdata\local\packages\pythonsoftwarefoundation.python.3.9_qbz5n2kfra8p0\localcache\local-packages\
python39\site-packages (from -r requirements.txt (line 3)) (1.10.0)
```

- Open the terminal.
- Run the command: pip install -r requirements.txt
- This command installs all the libraries listed in the requirements.txt file

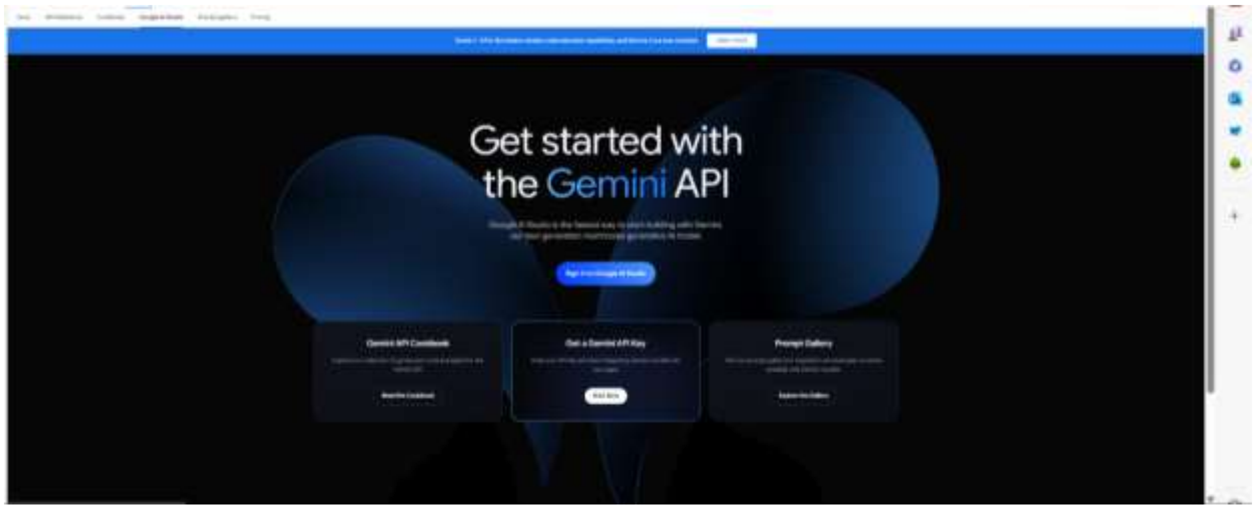
MILESTONE 2 : INITIALIZATION THE MODEL

The Google API key is a secure access token provided by Google, enabling developers to authenticate and interact with various Google APIs. It acts as a form of identification, allowing users to access specific Google services and resources. This key plays a crucial role in authorizing and securing API requests, ensuring that only authorized users can access and utilize Google's services.For initializing the model we need to generate PALM API.

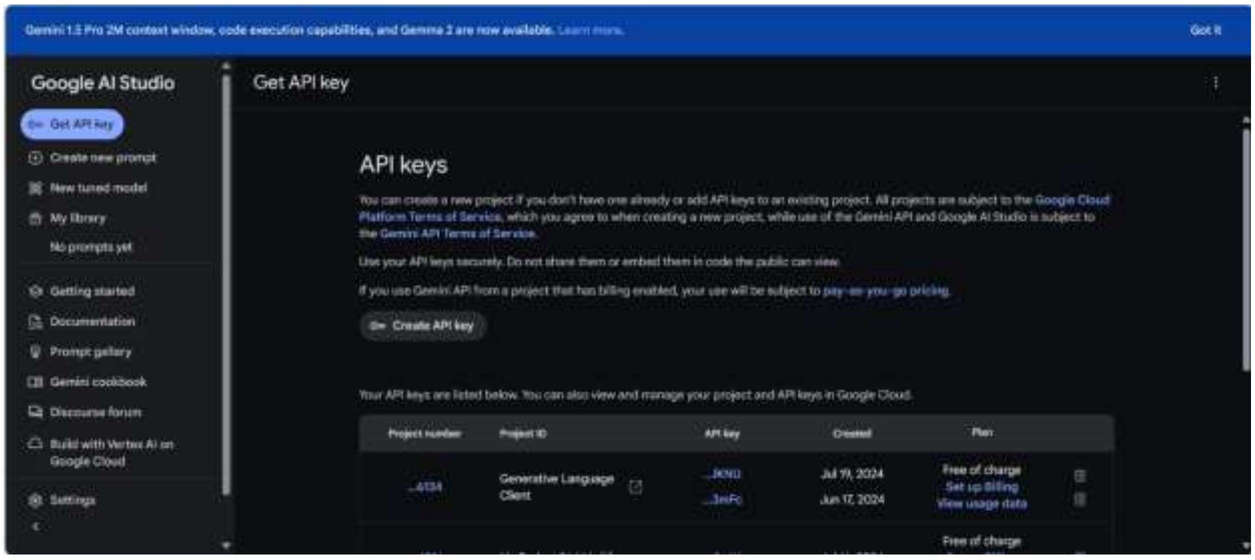
ACTIVITY 1 :

Generate PALM API Key

- Click on the link(https://ai.google.dev/aistudio).
- Then click on “Get API key in Google AI Studio”.
- Click on “Get API key” from the right navigation menu.
- Now click on “Create API key”. (Refer the below images)
- Copy the API key.



After signing in to your account, navigate to the 'Get an API Key' option. Clicking on this option will redirect you to another webpage as shown below.



Gemini 1.5 Pro 2M context window, code execution capabilities, and Gemma 2 are now available. Learn more.

Get API key

API keys

You can create a new project if you don't have one already or add API keys to an existing project. All projects are subject to the Google Cloud Platform Terms of Service, which you agree to when creating a new project, while use of the Gemini API and Google AI Studio is subject to the Gemini API Terms of Service.

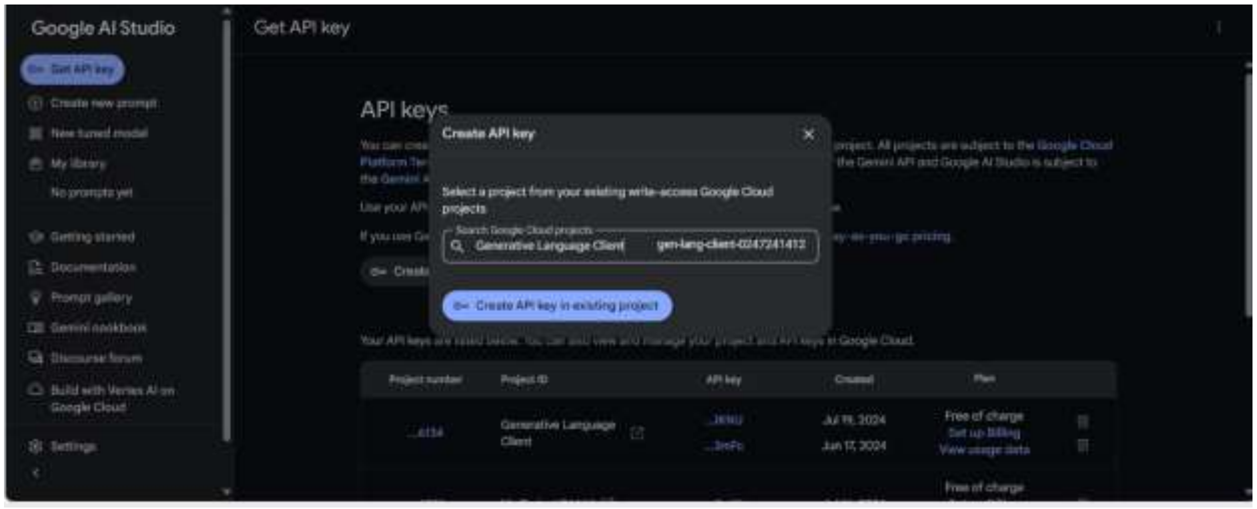
Use your API keys securely. Do not share them or embed them in code the public can view.

If you use Gemini API from a project that has billing enabled, your use will be subject to pay-as-you-go pricing.

Your API keys are listed below. You can also view and manage your project and API keys in Google Cloud.

Project number	Project ID	API key	Created	Plan
...4134	Generative Language Client	...JKNU ...3mFc	Jul 19, 2024 Jun 17, 2024	Free of charge Set up Billing View usage data
[illegible]	[illegible]	[illegible]	[illegible]	Free of charge

Next, click on 'Create API Key' and choose the generative language client as the project. Then, select 'Create API key in existing project'.



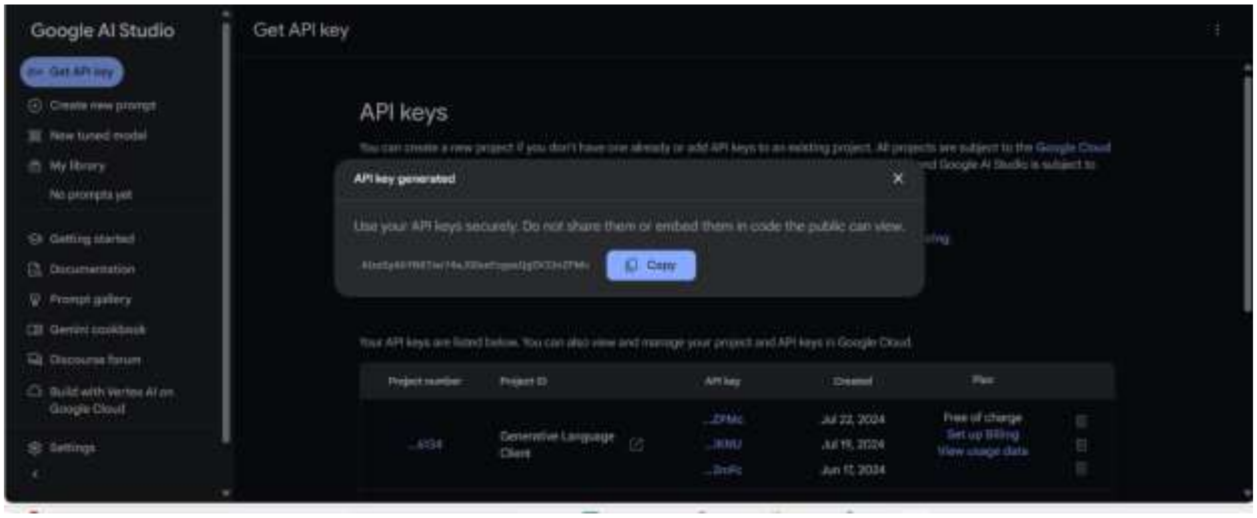
Select a project from your existing write-access Google Cloud projects

Search Google Cloud projects: Generative Language Client gen-lang-client-0247241412

Your API keys are listed below. You can also view and manage your project and API keys in Google Cloud.

Project number	Project ID	API key	Created	Plan
...6134	Generative Language Client	...JKNU	Jul 19, 2024	Free of charge Set up Billing View usage data
		...3nFc	Jun 17, 2024	
				Free of charge

Copy the newly generated API key as it is required for loading the pre-trained model.



You can create a new project if you don't have one already or add API keys to an existing project. All projects are subject to the Google Cloud ... and Google AI Studio is subject to

Use your API keys securely. Do not share them or embed them in code the public can view.

[illegible]

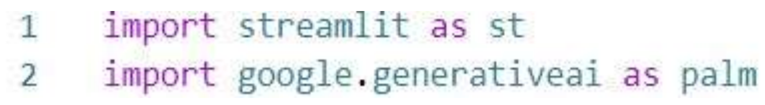
Your API keys are listed below. You can also view and manage your project and API keys in Google Cloud.

Project number	Project ID	API key	Created	Plan
...6134	Generative Language Client	...ZPMc	Jul 22, 2024	Free of charge Set up Billing View usage data
		...JKNU	Jul 19, 2024	
		...3nFc	Jun 17, 2024	

ACTIVITY 2 :

Initialize The Pre-Trained Model

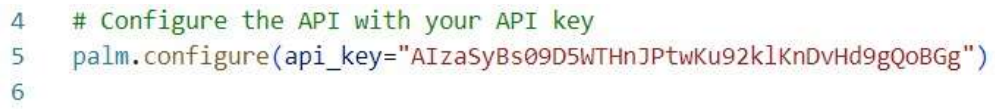
Import necessary files



```python
import streamlit as st
import google.generativeai as palm
```

- Streamlit, a popular Python library, is imported as st, enabling the creation of user interfaces directly within the Python script.
- Importing the palm module: This line imports the palm module from the google.generativeai package.

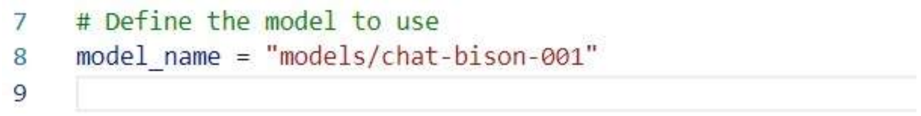
Configuration of the PALM API with the API key and initialize translator



```python
# Configure the API with your API key
palm.configure(api_key="AIzaSyBs09D5WTHnJPtwKu92klKnDvHd9gQoBGg")

```

- Configuring the API key: The configure function is used to set up or configure the Google API with an API key. The provided API key, in this case, is " AIzaSyBs09D5WTHnJPtwKu92klKnDvHd9gQoBGg".
- The Translator class facilitates language translation capabilities within the application.

Define the model to be used



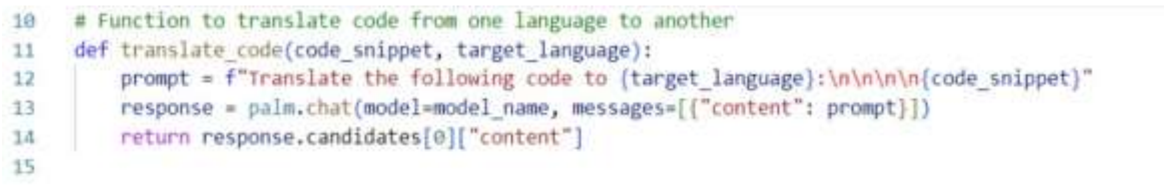
```python
# Define the model to use
model_name = "models/chat-bison-001"

```

- The line model_name = "models/chat-bison-001" sets the variable model_name to the string "models/chat-bison-001", which identifies a specific model provided by Google's PaLM (Pathways Language Model) API.
- This variable is used in subsequent API calls to specify which model to use for generating responses based on user prompts.

<u>MILESTONE 3</u> : INTERFACING WITH PRE-TRAINED MODEL

In this milestone, we will build a prompt template to generate code based on user description and language.

Function To Change The Programming Language Of The Code



```python
# Function to translate code from one language to another
def translate_code(code_snippet, target_language):
    prompt = f"Translate the following code to {target_language}:\n\n\n\n{code_snippet}"
    response = palm.chat(model=model_name, messages=[{"content": prompt}])
    return response.candidates[0]["content"]

```

- The translate_code function is designed to translate a given code snippet from one programming language to another. It constructs a prompt that specifies the target programming language for translation and includes the provided code snippet.
- This prompt is then sent to the Google PaLM API using the palm.chat function, which processes the request based on the specified model.

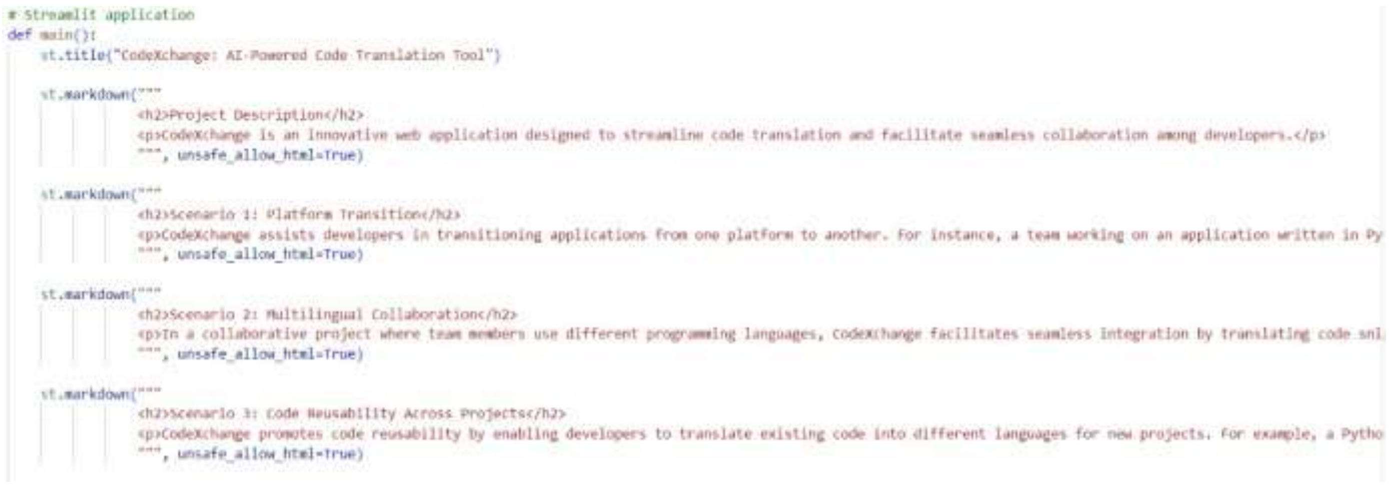
- Upon receiving a response, the function extracts and returns the content of the first candidate, which represents the translated code snippet.

<u>MILESTONE 4</u> : MODEL DEPLOYMENT

In this milestone, we are deploying the created model using streamlit. Model deployment using Streamlit involves creating a user-friendly web interface for deploying machine learning models, enabling users to interact with the model through a browser. Streamlit provides easy-to-use tools for developing and deploying data-driven applications, allowing for seamless integration of machine learning models into web-based applications.

ACTIVITY 1 :

Give The Project Title, Description And Describe The Scenarios



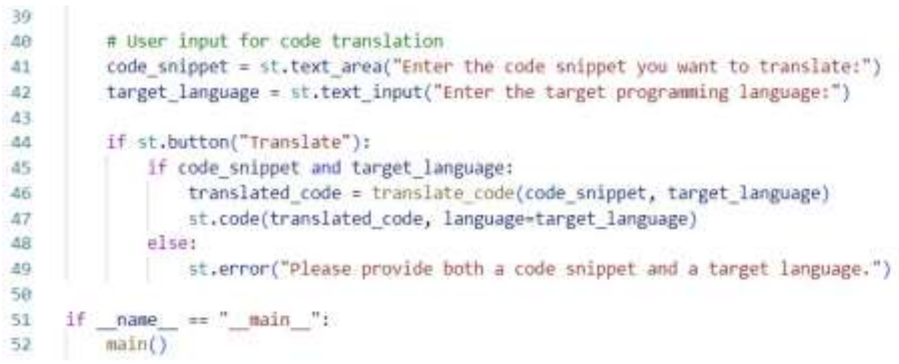
```
# Streamlit application
def main():
    st.title("CodeXchange: AI-Powered Code Translation Tool")

    st.markdown("""
            <h2>Project Description</h2>
            <p>CodeXchange is an innovative web application designed to streamline code translation and facilitate seamless collaboration among developers.</p>
            """, unsafe_allow_html=True)

    st.markdown("""
            <h2>Scenario 1: Platform Transition</h2>
            <p>CodeXchange assists developers in transitioning applications from one platform to another. For instance, a team working on an application written in Py
            """, unsafe_allow_html=True)

    st.markdown("""
            <h2>Scenario 2: Multilingual Collaboration</h2>
            <p>In a collaborative project where team members use different programming languages, CodeXchange facilitates seamless integration by translating code sni
            """, unsafe_allow_html=True)

    st.markdown("""
            <h2>Scenario 3: Code Reusability Across Projects</h2>
            <p>CodeXchange promotes code reusability by enabling developers to translate existing code into different languages for new projects. For example, a Pytho
            """, unsafe_allow_html=True)
```

- The main function serves as the entry point for the CodeXchange application, presenting a user interface that provides access to the tool's functionalities and features.

- It begins by setting the title of the application as "CodeXchange: AI-Powered Code Translation Tool" and provides a comprehensive project description using Markdown to explain the tool's purpose.
- The description highlights CodeXchange's ability to streamline code translation and facilitate collaboration among developers working with different programming languages.

ACTIVITY 2 :

Translate The Code Given By The User



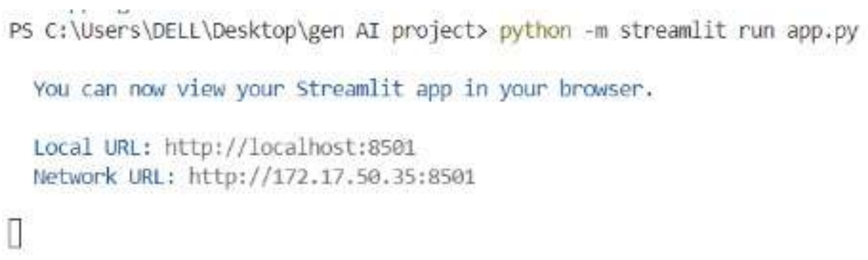
```python

    # User input for code translation
    code_snippet = st.text_area("Enter the code snippet you want to translate:")
    target_language = st.text_input("Enter the target programming language:")

    if st.button("Translate"):
        if code_snippet and target_language:
            translated_code = translate_code(code_snippet, target_language)
            st.code(translated_code, language=target_language)
        else:
            st.error("Please provide both a code snippet and a target language.")

if __name__ == "__main__":
    main()
```

- The subheader "Code Translation" is displayed, followed by a text area input field where the user can enter the code snippet they want to translate.
- Additionally, the user is asked to select the target programming language for the translation from a dropdown menu.
- When the user clicks the "Translate Code" button, the code attempts to translate the provided code snippet into the specified target language.
- If the user has entered a code snippet, it triggers the translation process. During translation, a spinner indicates that the code is being translated. If the translation process is successful, the translated code is displayed with syntax highlighting corresponding to the target language.

- If an error occurs during translation, an error message is shown.
- If the user has not entered a code snippet, they are prompted to do so.
- Finally, the main() function is called to execute the entire code when the script is run.

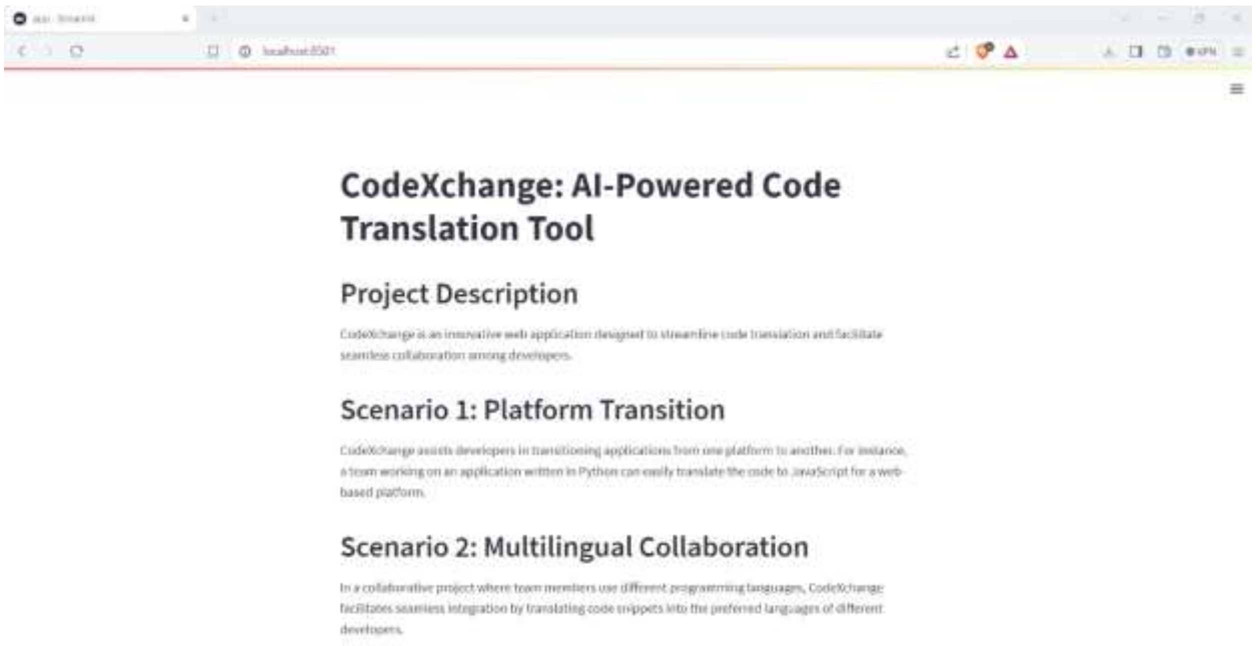
ACTIVITY 3 : Run The Web Application

- Open the anaconda prompt from the start menu
- Navigate to the folder where your Python script is.
- Now type “streamlit run app.py” command
- Navigate to the localhost where you can view your web page



```
PS C:\Users\DELL\Desktop\gen AI project> python -m streamlit run app.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://172.17.50.35:8501

```

Now, the application will open in the web browser.



localhost:8501

CodeXchange: AI-Powered Code Translation Tool

Project Description

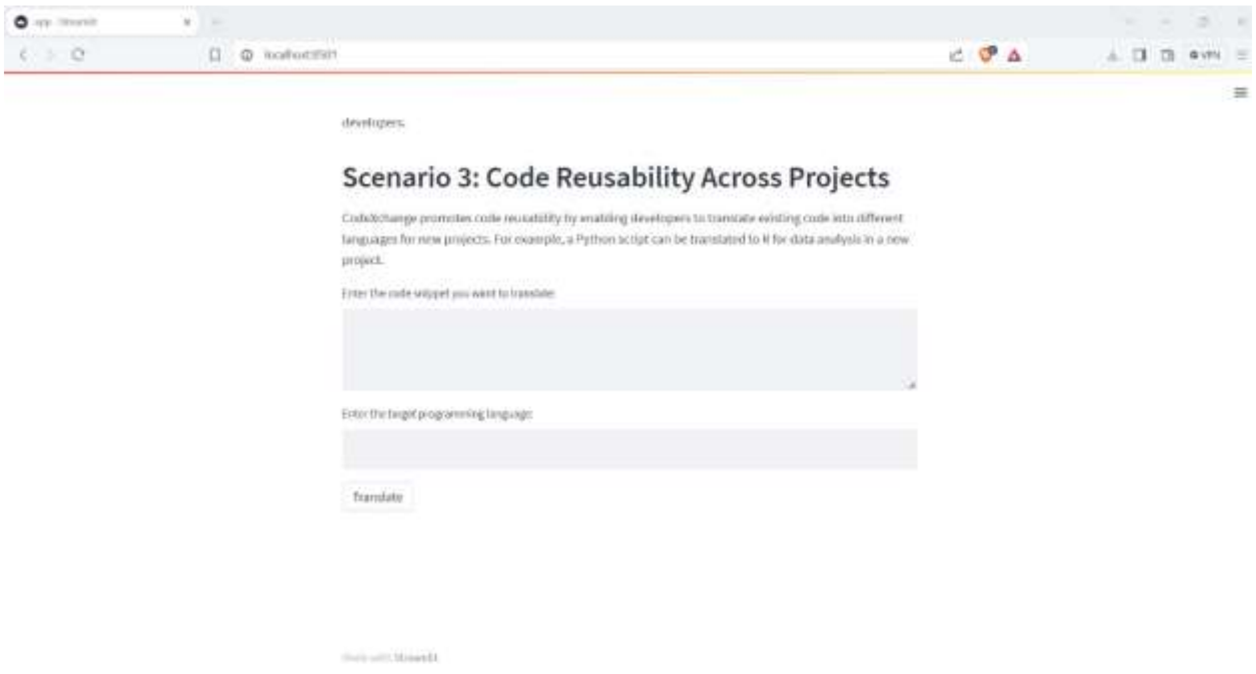
CodeXchange is an innovative web application designed to streamline code translation and facilitate seamless collaboration among developers.

Scenario 1: Platform Transition

CodeXchange assists developers in transitioning applications from one platform to another. For instance, a team working on an application written in Python can easily translate the code to JavaScript for a web-based platform.

Scenario 2: Multilingual Collaboration

In a collaborative project where team members use different programming languages, CodeXchange facilitates seamless integration by translating code snippets into the preferred languages of different developers.



localhost:8501

developers.

Scenario 3: Code Reusability Across Projects

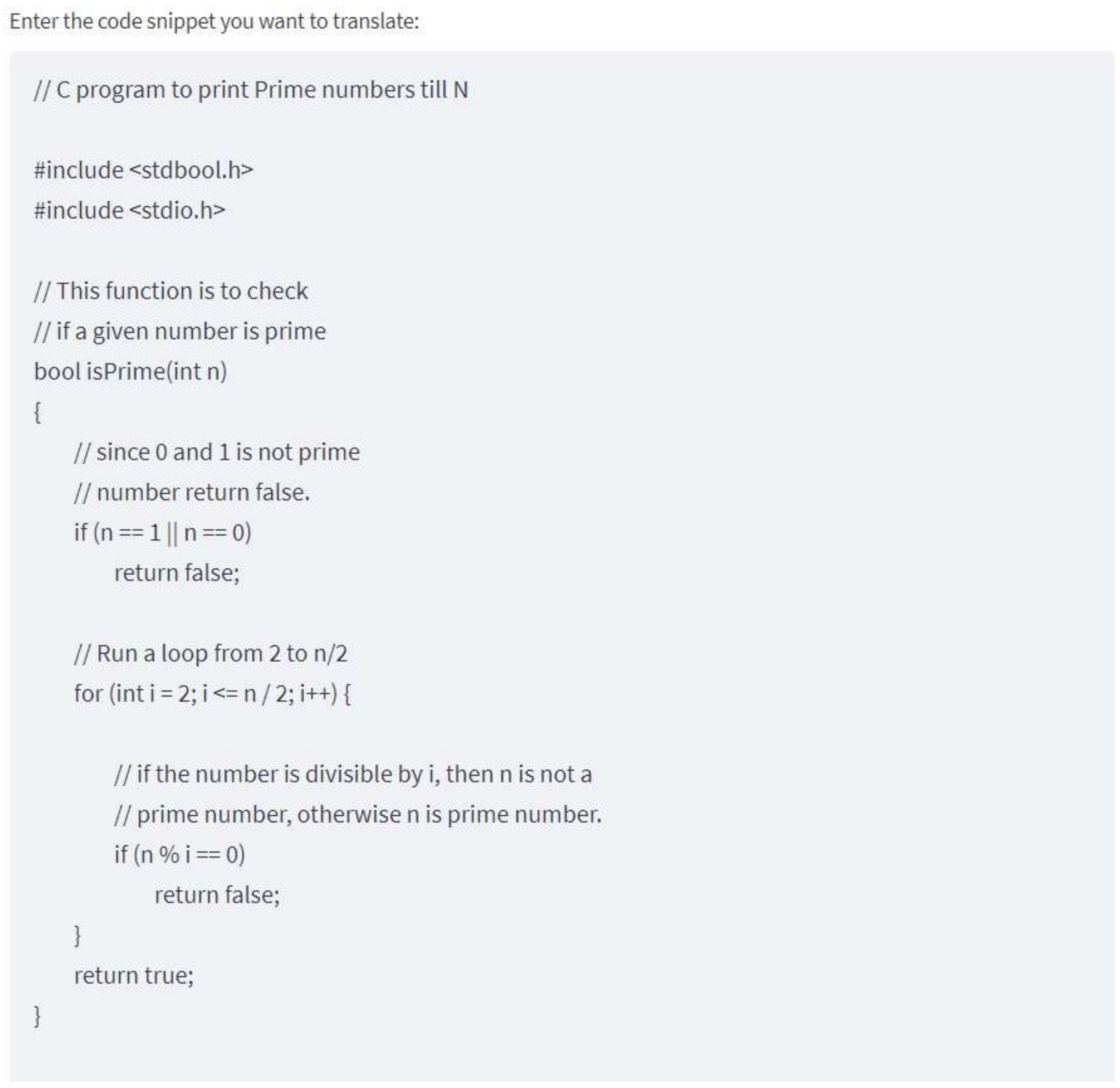
CodeXchange promotes code reusability by enabling developers to translate existing code into different languages for new projects. For example, a Python script can be translated to R for data analysis in a new project.

Enter the code snippet you want to translate:

Enter the target programming language:

Output 1: Translate C code to C++ code

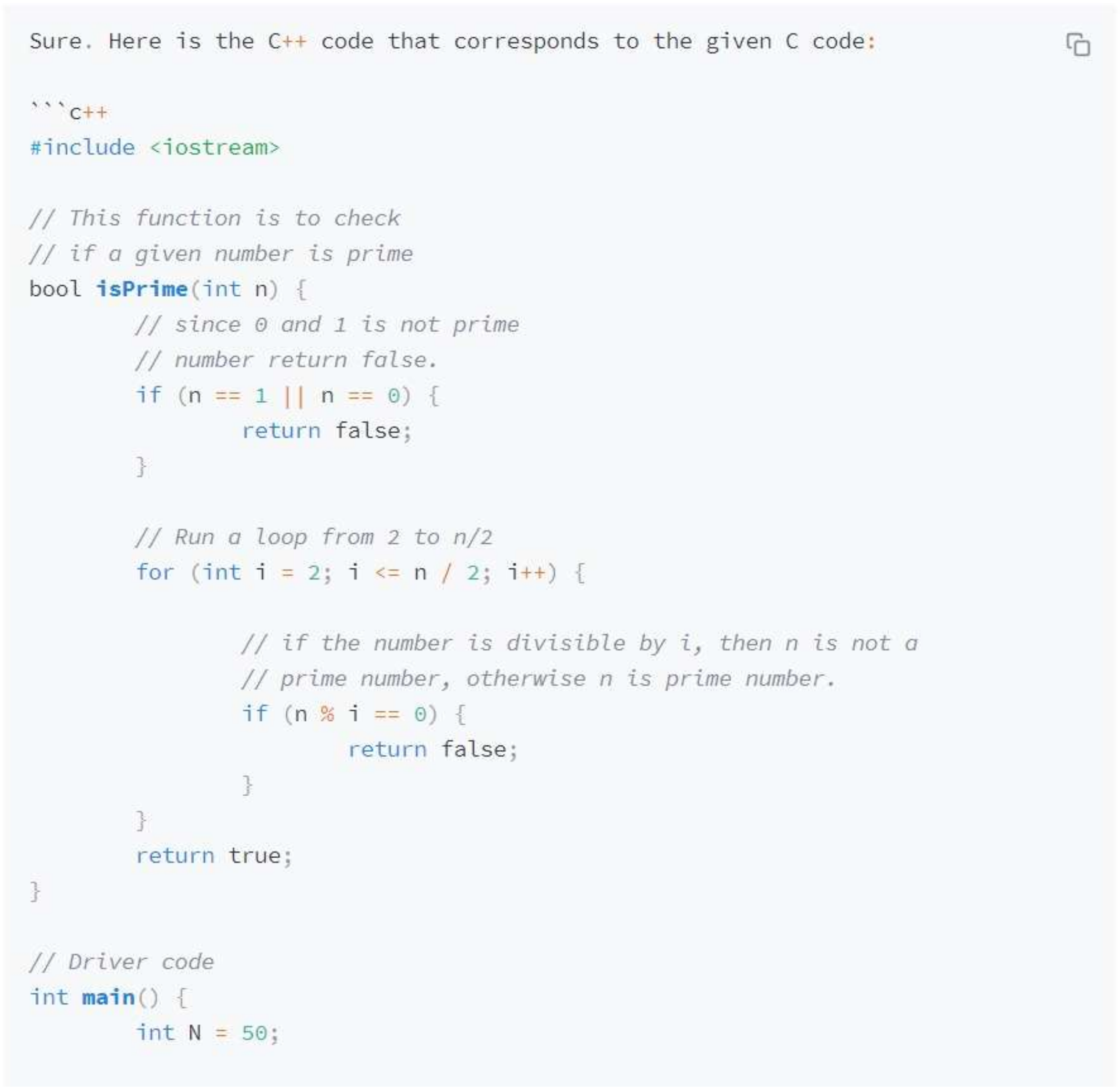
INPUT : C CODE



Enter the code snippet you want to translate:

```
// C program to print Prime numbers till N

#include <stdbool.h>
#include <stdio.h>

// This function is to check
// if a given number is prime
bool isPrime(int n)
{
    // since 0 and 1 is not prime
    // number return false.
    if (n == 1 || n == 0)
        return false;

    // Run a loop from 2 to n/2
    for (int i = 2; i <= n / 2; i++) {

        // if the number is divisible by i, then n is not a
        // prime number, otherwise n is prime number.
        if (n % i == 0)
            return false;
    }
    return true;
}
```

OUTPUT : C++



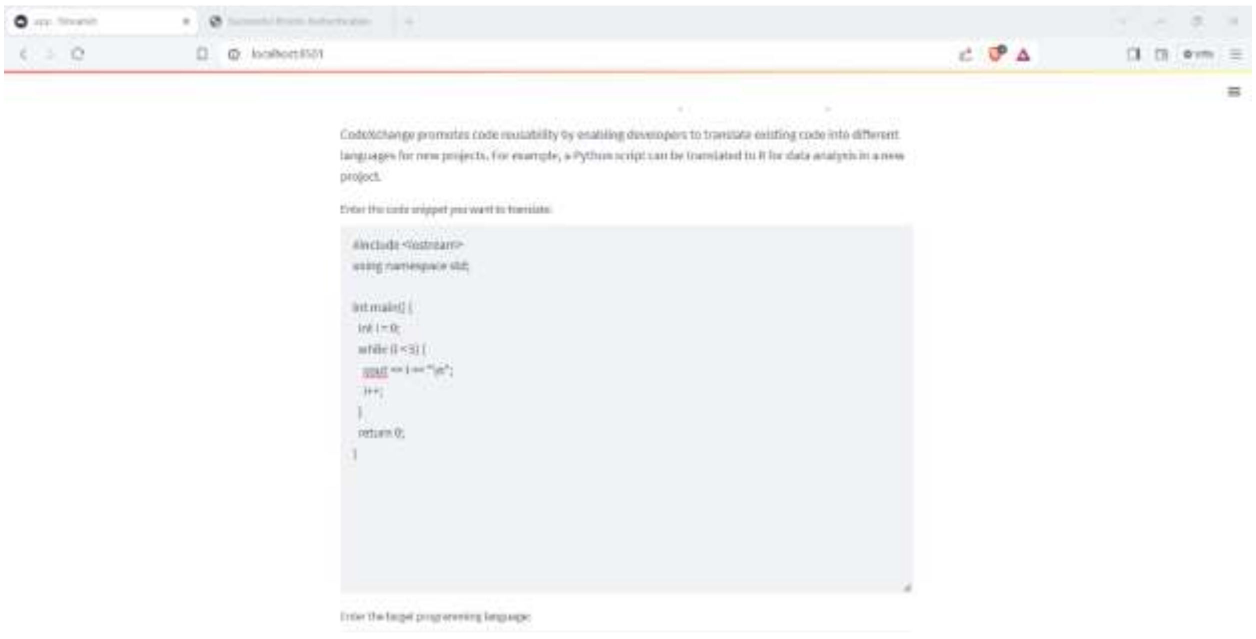
Sure. Here is the C++ code that corresponds to the given C code:

~~~
```c++
#include <iostream>

// This function is to check
// if a given number is prime
bool isPrime(int n) {
        // since 0 and 1 is not prime
        // number return false.
        if (n == 1 || n == 0) {
                return false;
        }

        // Run a loop from 2 to n/2
        for (int i = 2; i <= n / 2; i++) {

                // if the number is divisible by i, then n is not a
                // prime number, otherwise n is prime number.
                if (n % i == 0) {
                        return false;
                }
        }
        return true;
}

// Driver code
int main() {
        int N = 50;
~~~

```cpp
        }
        return true;
}

// Driver code
int main() {
        int N = 50;

        // check for the every number from 1 to N
        for (int i = 1; i <= N; i++) {

                // check if i (current number) is prime
                if (isPrime(i)) {
                        std::cout << i << " ";
                }
        }
        std::cout << std::endl;
        return 0;
}
```

Output1: Translate C++ code to java code

Input C++ code



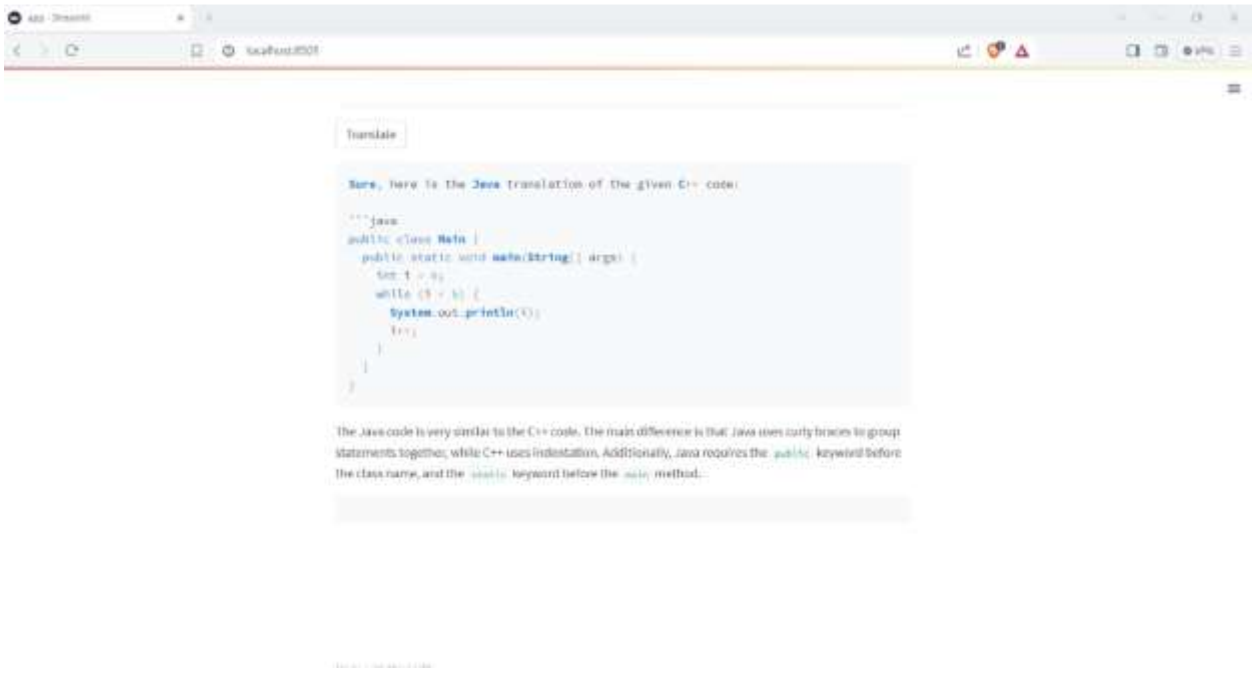
CodeXchange promotes code reusability by enabling developers to translate existing code into different languages for new projects. For example, a Python script can be translated to R for data analysis in a new project.

Enter the code snippet you want to translate:

```
#include <iostream>
using namespace std;

int main() {
  int i = 0;
  while (i < 3) {
   cout << i << "\n";
   i++;
  }
  return 0;
}
```

Enter the target programming language:

Output : java code

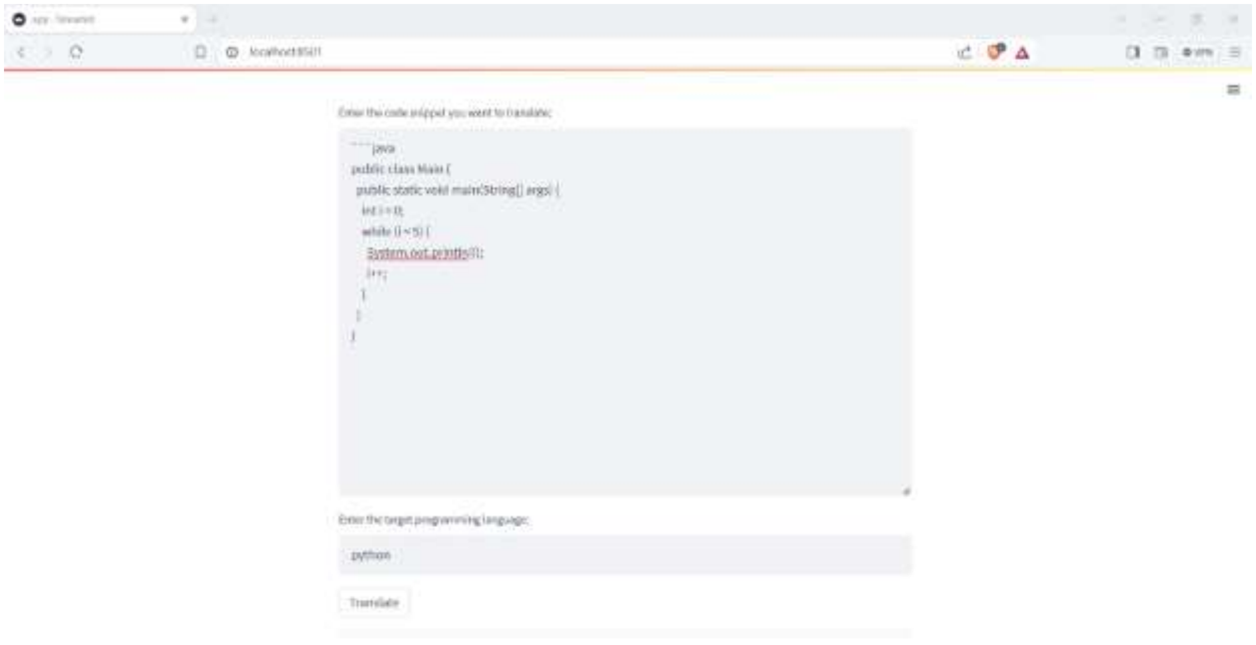


Translate

~~~
Sure, here is the Java translation of the given C++ code:

```java
public class Main {
  public static void main(String[] args) {
    int i = 0;
    while (i < 5) {
      System.out.println(i);
      i++;
    }
  }
}
~~~

The Java code is very similar to the C++ code. The main difference is that Java uses curly braces to group statements together, while C++ uses indentation. Additionally, Java requires the `public` keyword before the class name, and the `static` keyword before the `main` method.

Output1: Translate java code to python code



Enter the code snippet you want to translate:

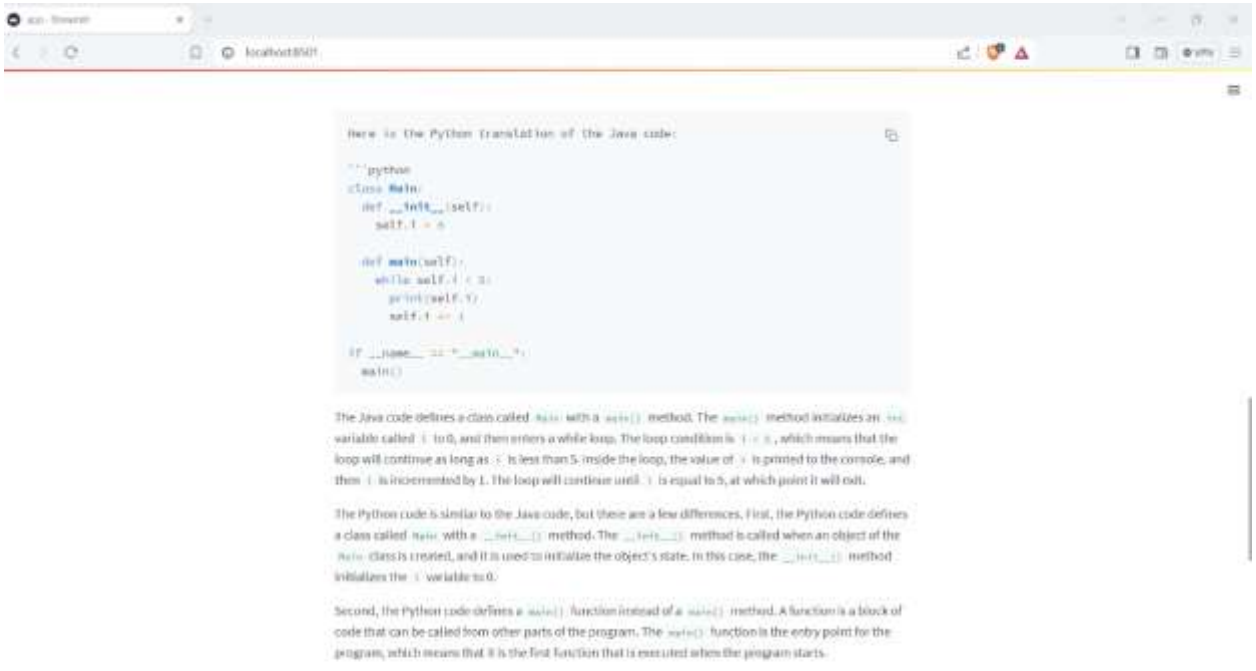
~~~
```java
public class Main {
  public static void main(String[] args) {
    int i = 0;
    while (i < 5) {
      System.out.println(i);
      i++;
    }
  }
}
~~~

Enter the target programming language:

python

Translate

Output: Python code



~~~
Here is the Python translation of the Java code:

```python
class Main:
  def __init__(self):
    self.i = 0

  def main(self):
    while self.i < 5:
      print(self.i)
      self.i += 1

if __name__ == "__main__":
  main()
~~~

The Java code defines a class called `Main` with a `main()` method. The `main()` method initializes an `int` variable called `i` to 0, and then enters a while loop. The loop condition is `i < 5`, which means that the loop will continue as long as `i` is less than 5. Inside the loop, the value of `i` is printed to the console, and then `i` is incremented by 1. The loop will continue until `i` is equal to 5, at which point it will exit.

The Python code is similar to the Java code, but there are a few differences. First, the Python code defines a class called `Main` with a `__init__()` method. The `__init__()` method is called when an object of the `Main` class is created, and it is used to initialize the object's state. In this case, the `__init__()` method initializes the `i` variable to 0.

Second, the Python code defines a `main()` function instead of a `main()` method. A function is a block of code that can be called from other parts of the program. The `main()` function is the entry point for the program, which means that it is the first function that is executed when the program starts.