CodeXchange: An AI-Powered Code Translator Tool Using Palm’s Text-Baison-001

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.

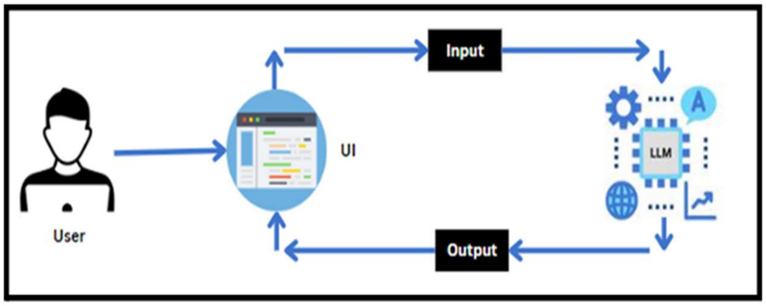
Scenario 2: Multilingual Collaboration

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 code based on the input text.
- The generated code are sent back to the frontend for display on the UI.
- Users can view the dynamically generated code 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
  - Code Generator
- Model Deployment
  - Deploy the application using Streamlit

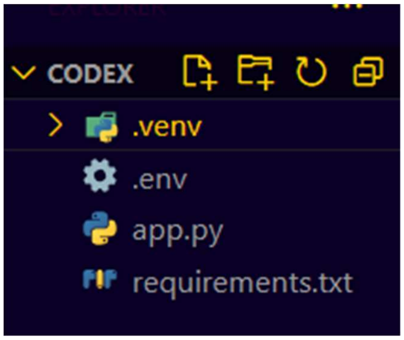
Prior Knowledge

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:

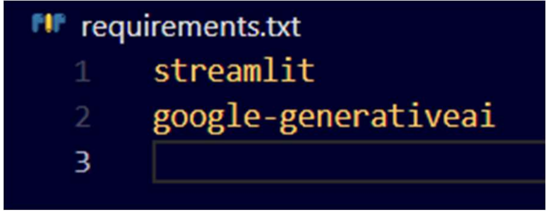


- app.py: It serves as the primary application file housing both the model and Streamlit UI code.
- .env file: It securely stores the Google API key.
- requirements.txt: It enumerates the libraries necessary for installation to ensure proper functioning.
- Additionally, ensure proper file organization and adhere to best practices for version control.

MILESTONE - 1 -> Requirements 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.

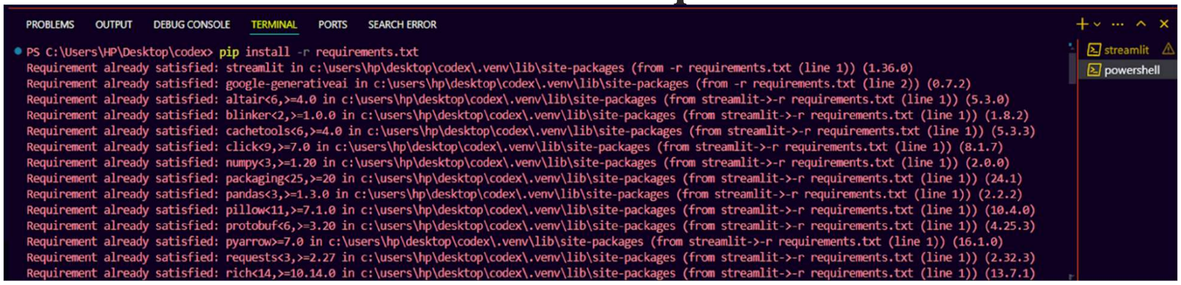


requirements.txt

```
streamlit
google-generativeai

```

- 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\HP\Desktop\codex> pip install -r requirements.txt
Requirement already satisfied: streamlit in c:\users\hp\desktop\codex\.venv\lib\site-packages (from -r requirements.txt (line 1)) (1.36.0)
Requirement already satisfied: google-generativeai in c:\users\hp\desktop\codex\.venv\lib\site-packages (from -r requirements.txt (line 2)) (0.7.2)
Requirement already satisfied: altair<6,>=4.0 in c:\users\hp\desktop\codex\.venv\lib\site-packages (from streamlit->-r requirements.txt (line 1)) (5.3.0)
Requirement already satisfied: blinker<2,>=1.0.0 in c:\users\hp\desktop\codex\.venv\lib\site-packages (from streamlit->-r requirements.txt (line 1)) (1.8.2)
Requirement already satisfied: cachetools<6,>=4.0 in c:\users\hp\desktop\codex\.venv\lib\site-packages (from streamlit->-r requirements.txt (line 1)) (5.3.3)
Requirement already satisfied: click<9,>=7.0 in c:\users\hp\desktop\codex\.venv\lib\site-packages (from streamlit->-r requirements.txt (line 1)) (8.1.7)
Requirement already satisfied: numpy<3,>=1.20 in c:\users\hp\desktop\codex\.venv\lib\site-packages (from streamlit->-r requirements.txt (line 1)) (2.0.0)
Requirement already satisfied: packaging<25,>=20 in c:\users\hp\desktop\codex\.venv\lib\site-packages (from streamlit->-r requirements.txt (line 1)) (24.1)
Requirement already satisfied: pandas<3,>=1.3.0 in c:\users\hp\desktop\codex\.venv\lib\site-packages (from streamlit->-r requirements.txt (line 1)) (2.2.2)
Requirement already satisfied: pillow<11,>=7.1.0 in c:\users\hp\desktop\codex\.venv\lib\site-packages (from streamlit->-r requirements.txt (line 1)) (10.4.0)
Requirement already satisfied: protobuf<6,>=3.20 in c:\users\hp\desktop\codex\.venv\lib\site-packages (from streamlit->-r requirements.txt (line 1)) (4.25.3)
Requirement already satisfied: pyarrow>=7.0 in c:\users\hp\desktop\codex\.venv\lib\site-packages (from streamlit->-r requirements.txt (line 1)) (16.1.0)
Requirement already satisfied: requests<3,>=2.27 in c:\users\hp\desktop\codex\.venv\lib\site-packages (from streamlit->-r requirements.txt (line 1)) (2.32.3)
Requirement already satisfied: rich<14,>=10.14.0 in c:\users\hp\desktop\codex\.venv\lib\site-packages (from streamlit->-r requirements.txt (line 1)) (13.7.1)
```

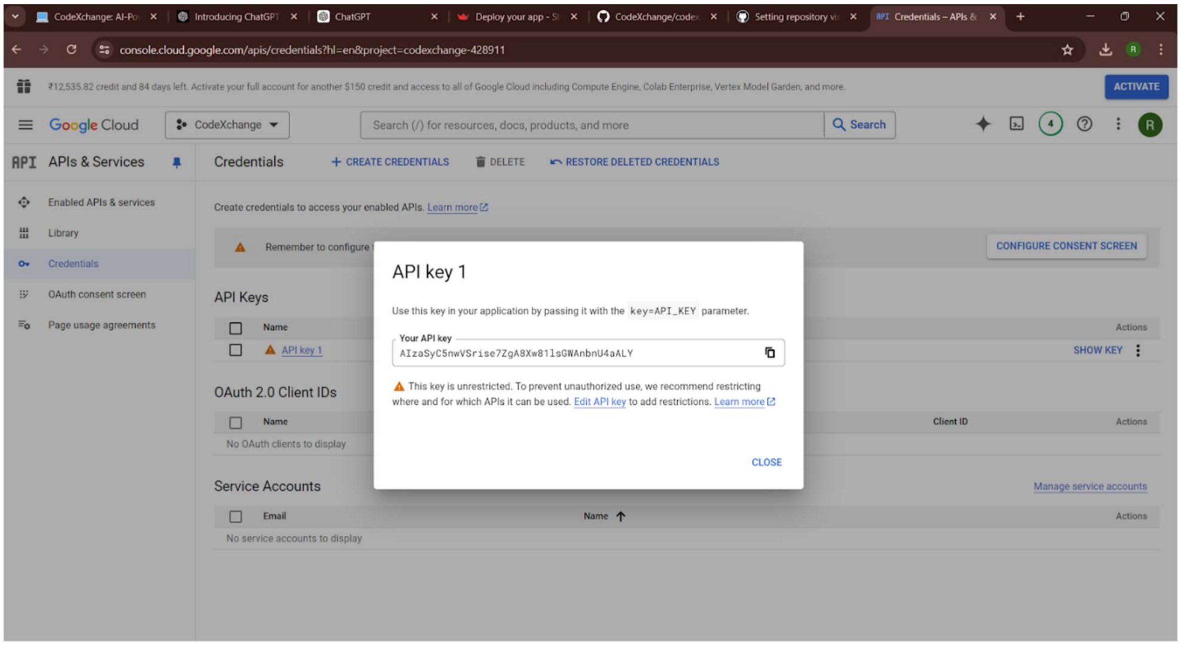
- Open the terminal.
- Run the command: pip install -r requirements.txt
- This command installs all the libraries listed in the requirements.txt file

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 the provided link to access the following webpage.
Link: https://console.cloud.google.com/welcome/new?project=<YOUR_PROJECT_ID>



console.cloud.google.com/apis/credentials?hl=en&project=codexchange-428911

₹12,535.82 credit and 84 days left. Activate your full account for another $150 credit and access to all of Google Cloud including Compute Engine, Colab Enterprise, Vertex Model Garden, and more.

Credentials

Create credentials to access your enabled APIs. Learn more

Remember to configure t

API Keys

Name	Actions
API key 1	SHOW KEY

OAuth 2.0 Client IDs

Name	Client ID	Actions
No OAuth clients to display		

Service Accounts

Email	Name	Actions
No service accounts to display		

API key 1

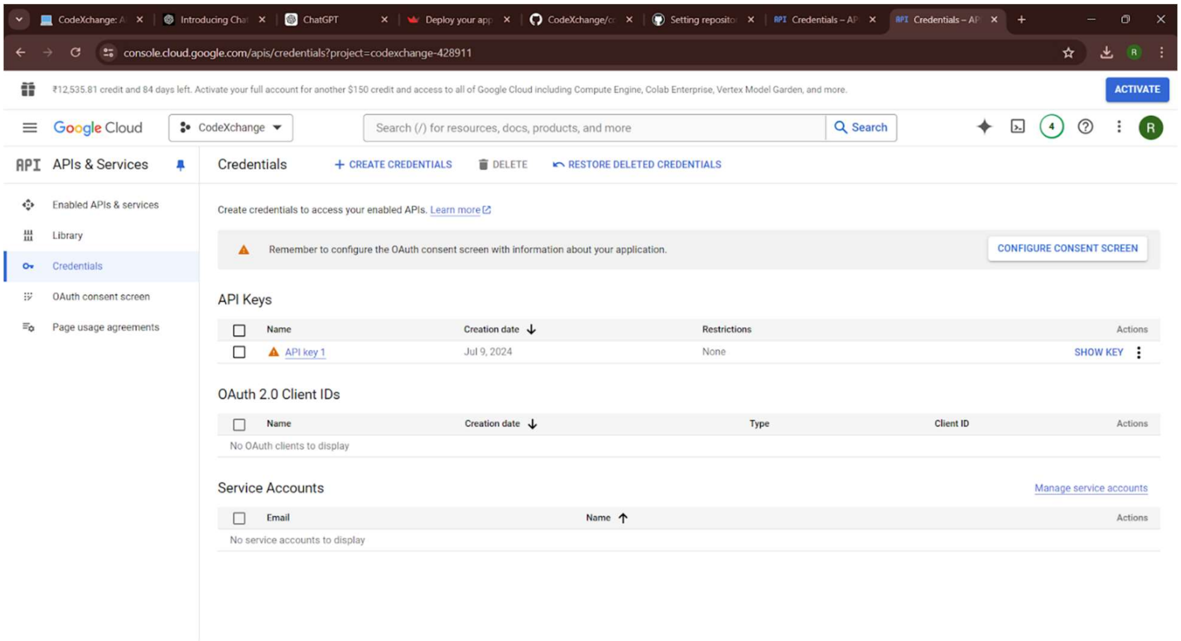
Use this key in your application by passing it with the `key=API_KEY` parameter.

Your API key

```
AIzaSyC5nwVSrise7ZgA8Xw81lsGWAnbnU4aALY
```

This key is unrestricted. To prevent unauthorized use, we recommend restricting where and for which APIs it can be used. Edit API key to add restrictions. Learn more

After signing in to your account, navigate to the API’s & Services > Credentials option. Clicking on this option will redirect you to another webpage as shown below.



console.cloud.google.com/apis/credentials?project=codexchange-428911

₹12,535.81 credit and 84 days left. Activate your full account for another $150 credit and access to all of Google Cloud including Compute Engine, Colab Enterprise, Vertex Model Garden, and more.

Credentials

Create credentials to access your enabled APIs. Learn more

Remember to configure the OAuth consent screen with information about your application.

API Keys

Name	Creation date ↓	Restrictions	Actions
API key 1	Jul 9, 2024	None	SHOW KEY

OAuth 2.0 Client IDs

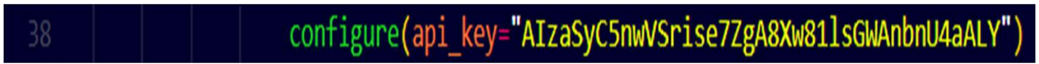
Name	Creation date ↓	Type	Client ID	Actions
No OAuth clients to display				

Service Accounts

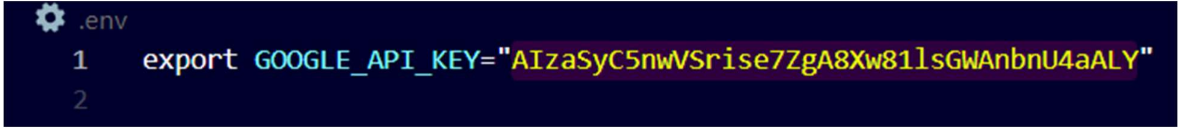
Email	Name ↑	Actions
No service accounts to display		

Next, click on 'Create API Key' under ‘Create Credentials’ and choose the generative language client as the project. Then, select 'Create API key in existing project'. Copy the newly generated API key as it is required for loading the PALM pre-trained model.

ACTIVITY - 2 -> Initialize the Pre-Trained API Key



```
configure(api_key="AIzaSyC5nwVSrise7ZgA8Xw81lsGWAnbnU4aALY")
```



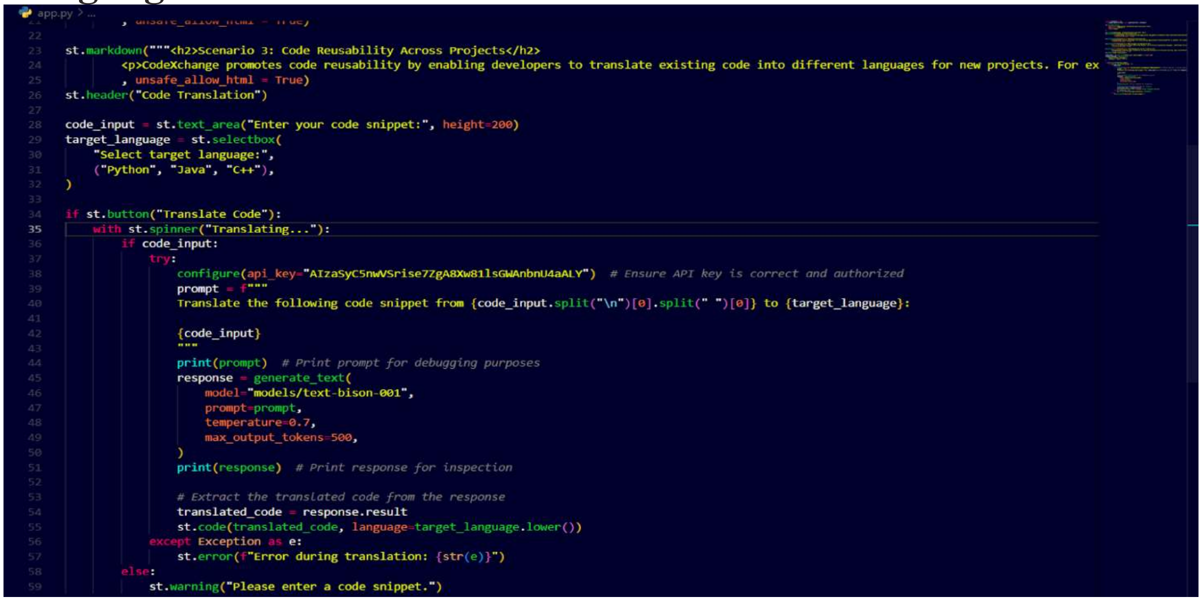
```
export GOOGLE_API_KEY="AIzaSyC5nwVSrise7ZgA8Xw81lsGWAnbnU4aALY"
```

- Create a .env file and define a variable named GOOGLE_API_KEY.
- Assign the copied Google API key to this variable.
- Paste the API key obtained from the previous steps here.

MILESTONE - 3 -> Interfacing With Pre-Trained Model

In this milestone, we will build a prompt template to generate code based on user description and language. To interface with the pre-trained model, we'll start by creating an app.py file, which will contain both the model and Streamlit UI code.

ACTIVITY - 1 -> Function To Change the Programming Language Of The Code



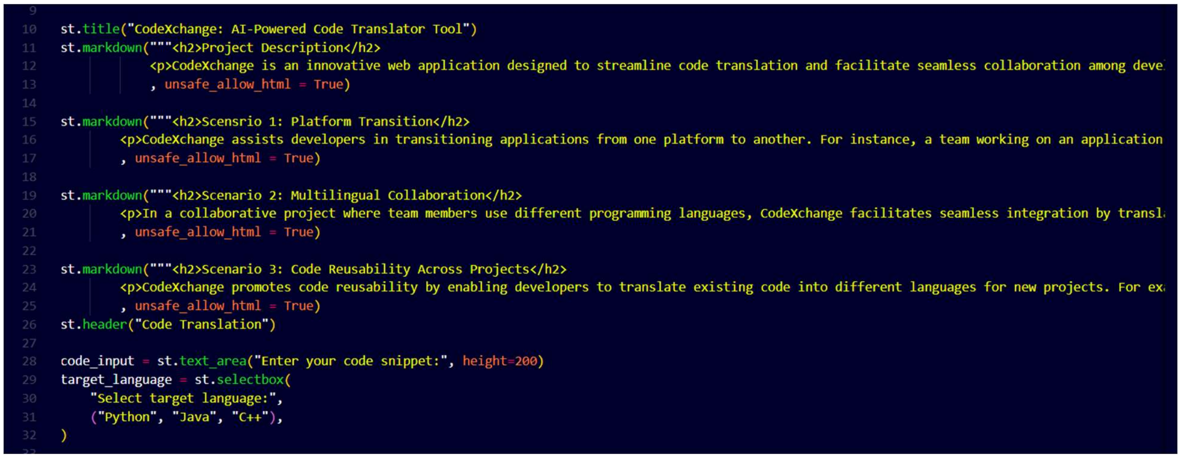
```
        , unsafe_allow_html = True)

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 ex
        , unsafe_allow_html = True)
st.header("Code Translation")

code_input = st.text_area("Enter your code snippet:", height=200)
target_language = st.selectbox(
    "Select target language:",
    ("Python", "Java", "C++"),
)

if st.button("Translate Code"):
    with st.spinner("Translating..."):
        if code_input:
            try:
                configure(api_key="AIzaSyC5nwVSrise7ZgA8Xw81lsGWAnbnU4aALY")  # Ensure API key is correct and authorized
                prompt = f"""
                Translate the following code snippet from {code_input.split("\n")[0].split(" ")[0]} to {target_language}:

                {code_input}
                """
                print(prompt)  # Print prompt for debugging purposes
                response = generate_text(
                    model="models/text-bison-001",
                    prompt=prompt,
                    temperature=0.7,
                    max_output_tokens=500,
                )
                print(response)  # Print response for inspection

                # Extract the translated code from the response
                translated_code = response.result
                st.code(translated_code, language=target_language.lower())
            except Exception as e:
                st.error(f"Error during translation: {str(e)}")
        else:
            st.warning("Please enter a code snippet.")
```

- The st_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 generate_text 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.

MILESTONE - 4 ->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.

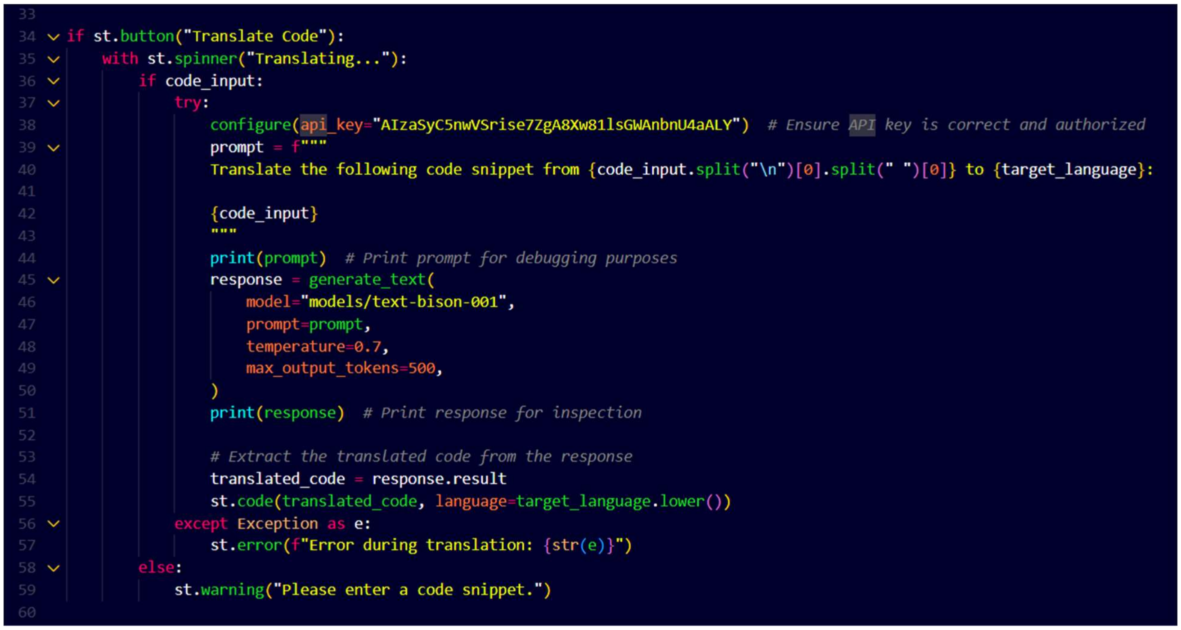
ACTIVITY - 1 -> Give The Project Title, Description And Describe The Scenarios



```
st.title("CodeXchange: AI-Powered Code Translator Tool")
st.markdown("""<h2>Project Description</h2>
            <p>CodeXchange is an innovative web application designed to streamline code translation and facilitate seamless collaboration among deve
            , unsafe_allow_html = True)

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

code_input = st.text_area("Enter your code snippet:", height=200)
target_language = st.selectbox(
    "Select target language:",
    ("Python", "Java", "C++"),
)
```

- 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 Translator 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

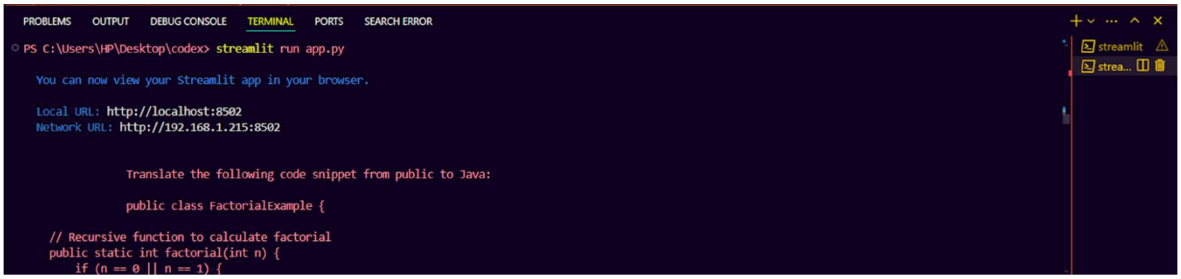


```python

if st.button("Translate Code"):
    with st.spinner("Translating..."):
        if code_input:
            try:
                configure(api_key="AIzaSyC5nwVSrise7ZgA8Xw81lsGWAnbnU4aALY")  # Ensure API key is correct and authorized
                prompt = f"""
                Translate the following code snippet from {code_input.split("\n")[0].split(" ")[0]} to {target_language}:

                {code_input}
                """
                print(prompt)  # Print prompt for debugging purposes
                response = generate_text(
                    model="models/text-bison-001",
                    prompt=prompt,
                    temperature=0.7,
                    max_output_tokens=500,
                )
                print(response)  # Print response for inspection

                # Extract the translated code from the response
                translated_code = response.result
                st.code(translated_code, language=target_language.lower())
            except Exception as e:
                st.error(f"Error during translation: {str(e)}")
        else:
            st.warning("Please enter a code snippet.")

```

- 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.

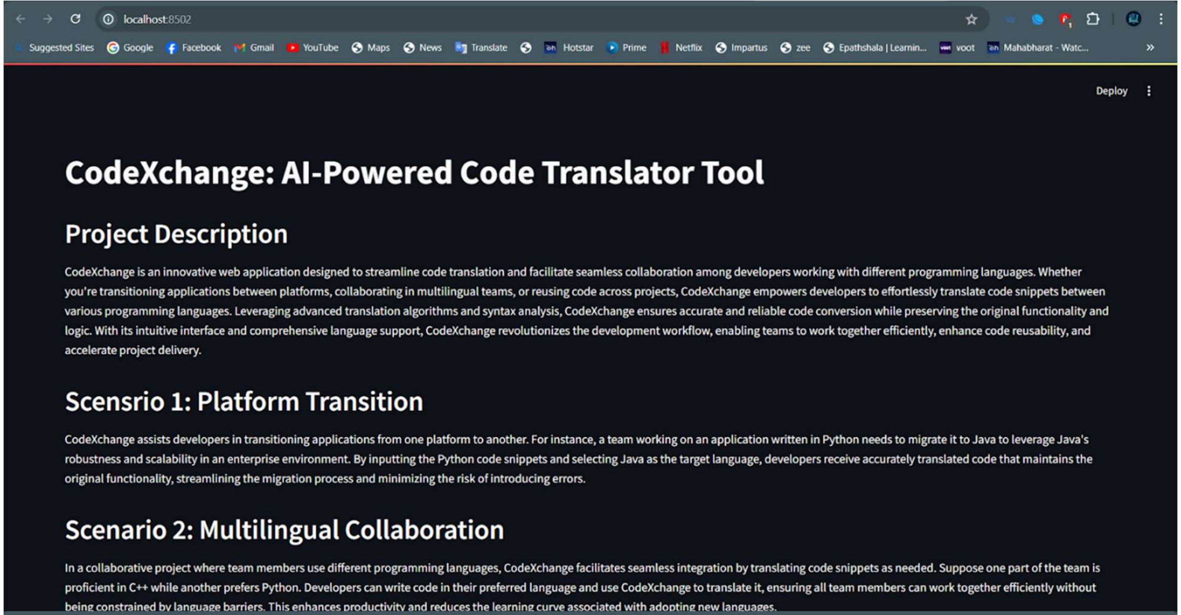
ACTIVITY - 3 -> Run The Web Application

- 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\HP\Desktop\codex> streamlit run app.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8502
  Network URL: http://192.168.1.215:8502


                Translate the following code snippet from public to Java:

                public class FactorialExample {

    // Recursive function to calculate factorial
    public static int factorial(int n) {
        if (n == 0 || n == 1) {
```

Now, the application will open in the web browser.



CodeXchange: AI-Powered Code Translator Tool

Project Description

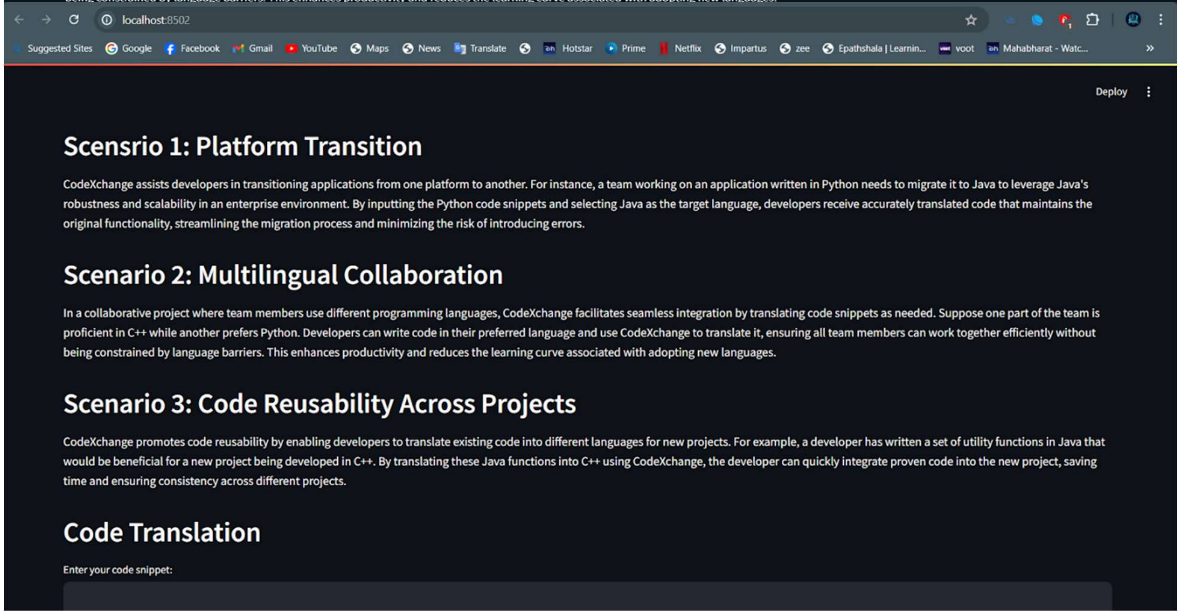
CodeXchange is an innovative web application designed to streamline code translation and facilitate seamless collaboration among developers working with different programming languages. Whether you're transitioning applications between platforms, collaborating in multilingual teams, or reusing code across 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.

Scensrio 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.



Scensrio 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

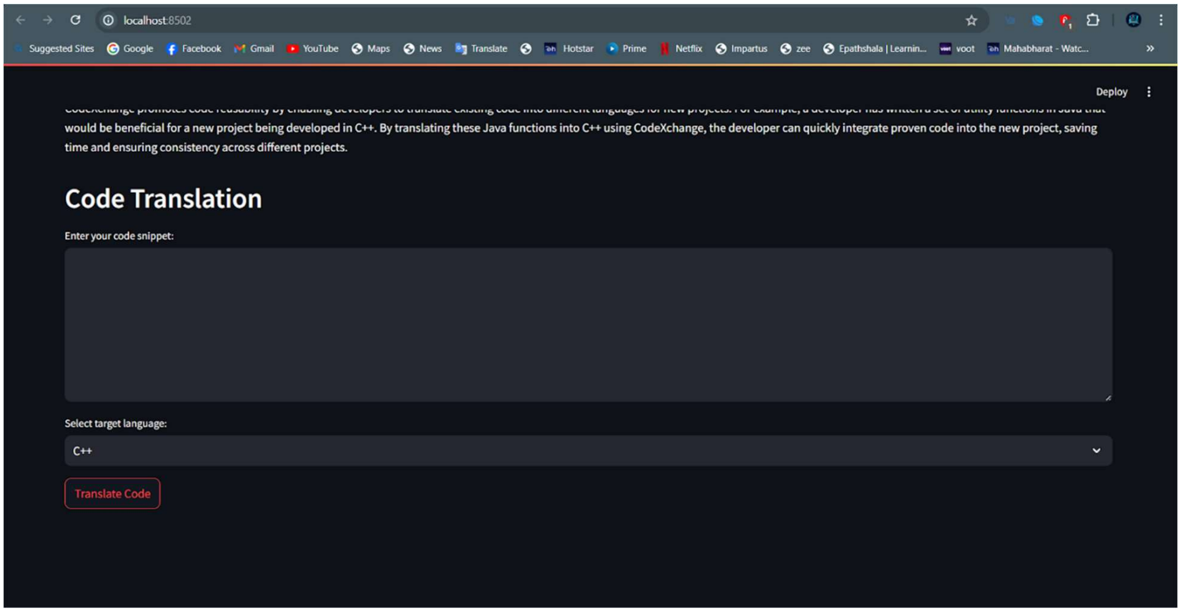
In a collaborative project where team members use different programming languages, CodeXchange facilitates seamless integration by translating code snippets as needed. Suppose one part of the team is proficient in C++ while another prefers Python. Developers can write code in their preferred language 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.

Code Translation

Enter your code snippet:



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.

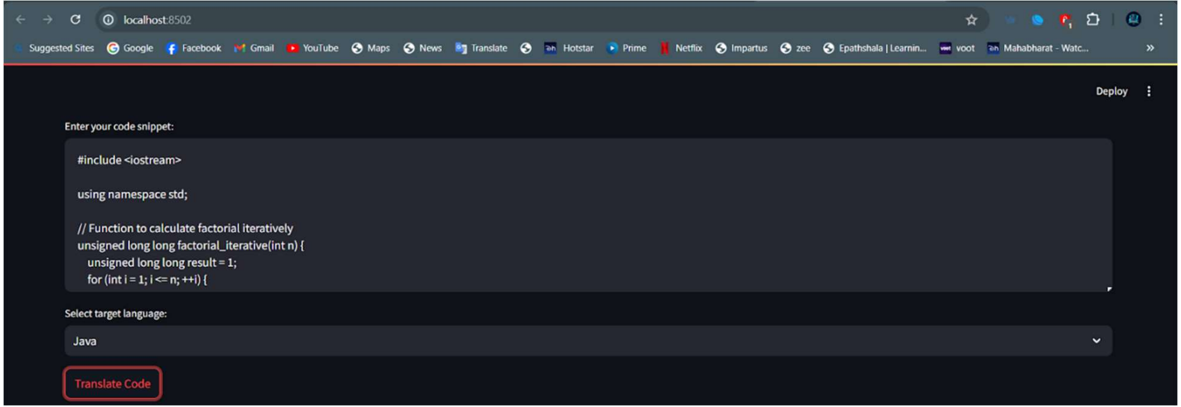
Code Translation

Enter your code snippet:

Select target language:

C++

Output 1: Translate C++ code to Java code

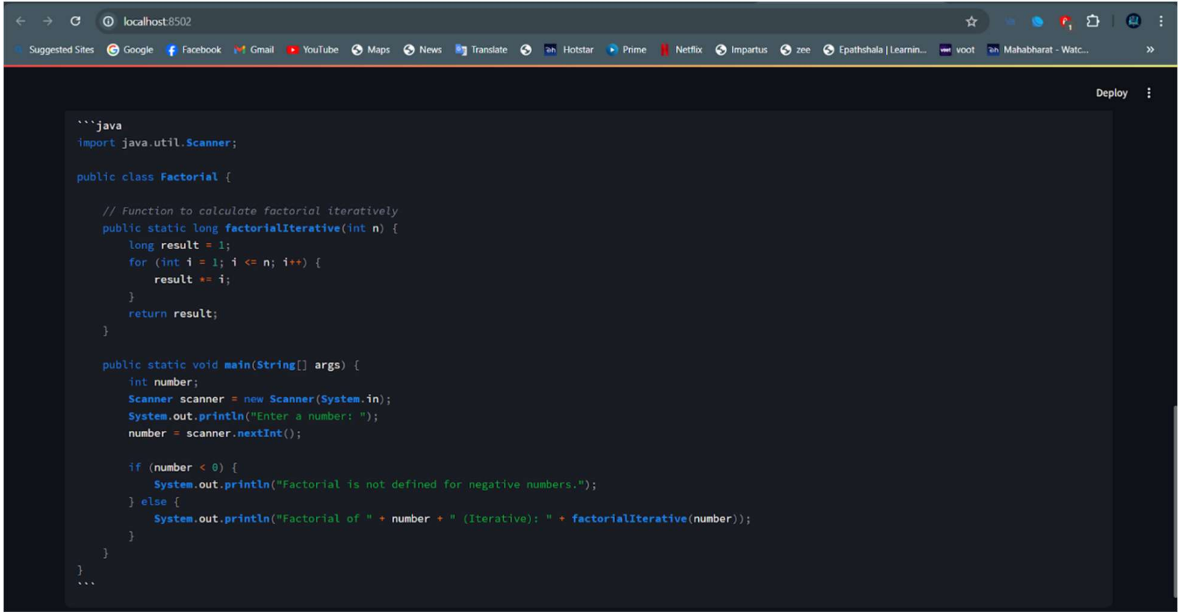


Enter your code snippet:

```
#include <iostream>

using namespace std;

// Function to calculate factorial iteratively
unsigned long long factorial_iterative(int n) {
   unsigned long long result = 1;
   for (int i = 1; i <= n; ++i) {
```

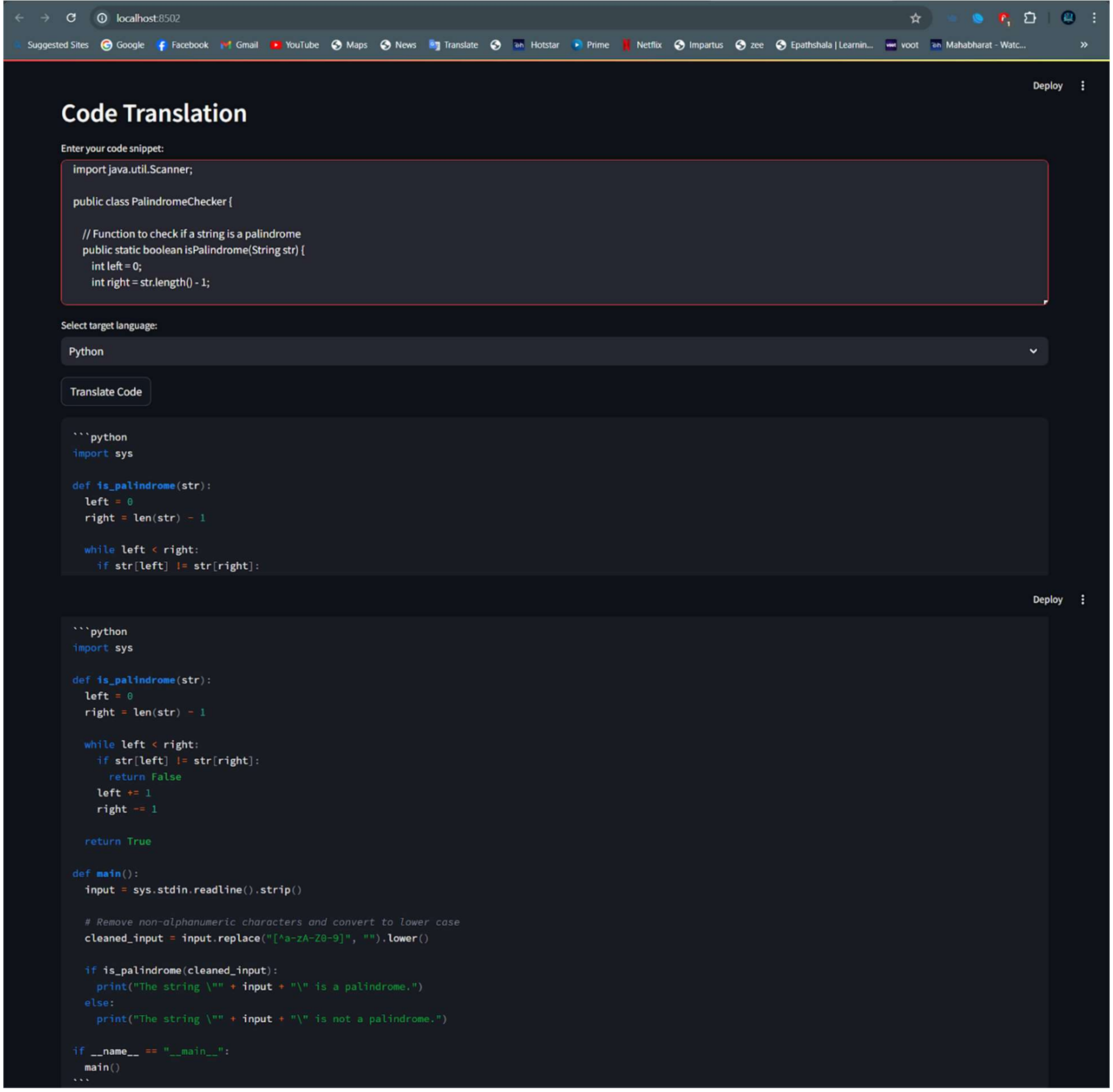
Select target language:

Java



~~~
```java
import java.util.Scanner;

public class Factorial {

    // Function to calculate factorial iteratively
    public static long factorialIterative(int n) {
        long result = 1;
        for (int i = 1; i <= n; i++) {
            result *= i;
        }
        return result;
    }

    public static void main(String[] args) {
        int number;
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter a number: ");
        number = scanner.nextInt();

        if (number < 0) {
            System.out.println("Factorial is not defined for negative numbers.");
        } else {
            System.out.println("Factorial of " + number + " (Iterative): " + factorialIterative(number));
        }
    }
}
```
~~~

Output 2: Java Code to Python



Code Translation

Enter your code snippet:

```
import java.util.Scanner;

public class PalindromeChecker {

  // Function to check if a string is a palindrome
  public static boolean isPalindrome(String str) {
    int left = 0;
    int right = str.length() - 1;
```

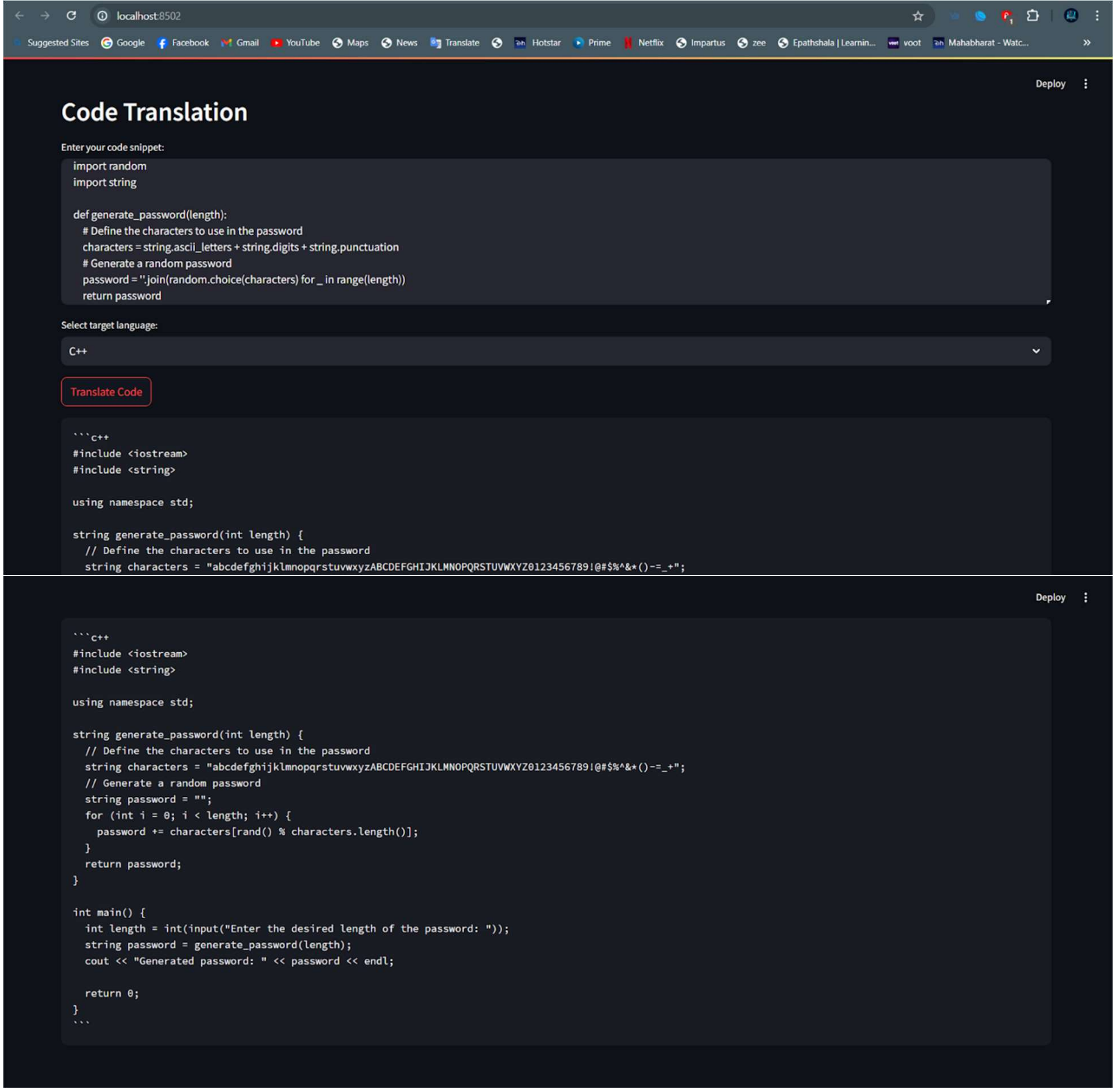
Select target language:

Python

~~~
```python
import sys

def is_palindrome(str):
  left = 0
  right = len(str) - 1

  while left < right:
    if str[left] != str[right]:
~~~

~~~
```python
import sys

def is_palindrome(str):
  left = 0
  right = len(str) - 1

  while left < right:
    if str[left] != str[right]:
      return False
    left += 1
    right -= 1

  return True

def main():
  input = sys.stdin.readline().strip()

  # Remove non-alphanumeric characters and convert to lower case
  cleaned_input = input.replace("[^a-zA-Z0-9]", "").lower()

  if is_palindrome(cleaned_input):
    print("The string \"" + input + "\" is a palindrome.")
  else:
    print("The string \"" + input + "\" is not a palindrome.")

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

Output 3: Python code to C++



Code Translation

Enter your code snippet:

```
import random
import string

def generate_password(length):
  # Define the characters to use in the password
  characters = string.ascii_letters + string.digits + string.punctuation
  # Generate a random password
  password = ''.join(random.choice(characters) for _ in range(length))
  return password
```

Select target language: C++

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

using namespace std;

string generate_password(int length) {
  // Define the characters to use in the password
  string characters = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-=_+";
~~~

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

using namespace std;

string generate_password(int length) {
  // Define the characters to use in the password
  string characters = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-=_+";
  // Generate a random password
  string password = "";
  for (int i = 0; i < length; i++) {
    password += characters[rand() % characters.length()];
  }
  return password;
}

int main() {
  int length = int(input("Enter the desired length of the password: "));
  string password = generate_password(length);
  cout << "Generated password: " << password << endl;

  return 0;
}
```
~~~