Genai –23BDS0014 Project

File For Screenshot Of Code and Output

CODE:

import streamlit as st

import google.generativeai as genai

import base64

genai.configure(api\_key="AIzaSyBP3\_R5mYhJJyBgSmTKWzmUYwoyV2LLmeI")

model = genai.GenerativeModel("gemini-1.5-flash")

st.set\_page\_config(page\_title="GenAI Code Assistant", page\_icon="💻")

st.markdown("""

    <style>

        .nav-tabs {

            display: flex;

            justify-content: center;

            margin-bottom: 20px;

        }

        .nav-tabs a {

            margin: 0 15px;

            padding: 10px 25px;

            text-decoration: none;

            color: white;

            background-color: #4B8BBE;

            border-radius: 10px;

            font-weight: bold;

        }

        .nav-tabs a.selected {

            background-color: #306998;

        }

    </style>

""", unsafe\_allow\_html=True)

tab = st.radio("Choose Tool:", ["🧠 Code Explainer", "🛠️ Code Debugger"], horizontal=True, label\_visibility="collapsed")

st.markdown(f"""

    <h1 style='text-align: center; color: #4B8BBE;'>{"🧠 GenAI Code Explainer" if tab == "🧠 Code Explainer" else "🛠️ GenAI Code Debugger"}</h1>

    <p style='text-align: center;'>Powered by Gemini · Supports Python, JavaScript, C++, and Java</p>

    <hr style="border: 1px solid #ccc;" />

""", unsafe\_allow\_html=True)

with st.sidebar:

    st.title("📌 About")

    st.markdown("""

        \*\*This GenAI App Helps You:\*\*

        - 🔍 Understand code in plain English

        - 🛠️ Debug and fix programming errors

        \*\*Languages Supported:\*\*

        - Python

        - JavaScript

        - C++

        - Java

        \*\*Made by:\*\* Krishna Gupta

        [🌐 GitHub](https://github.com/krishnagupta-hub1/ai-code-exp)

    """)

language = st.selectbox("🔤 Select Programming Language:", ["Python", "JavaScript", "C++", "Java"])

code\_input = st.text\_area(f"💻 Paste your {language} code here:", height=200)

def get\_download\_link(text):

    b64 = base64.b64encode(text.encode()).decode()

    return f'<a href="data:file/txt;base64,{b64}" download="output.txt">📥 Download Result</a>'

if tab == "🧠 Code Explainer":

    if st.button("✨ Explain Code"):

        if not code\_input.strip():

            st.warning("Please paste some code first.")

        else:

            with st.spinner("🧠 Analyzing your code..."):

                prompt = f"Explain the following {language} code in beginner-friendly English:\n\n{code\_input}"

                try:

                    response = model.generate\_content(prompt)

                    explanation = response.text

                    st.markdown("### 📝 Explanation:")

                    st.success(explanation)

                    st.markdown("### 📄 Your Code:")

                    st.code(code\_input, language=language.lower())

                    st.markdown(get\_download\_link(explanation), unsafe\_allow\_html=True)

                    st.balloons()

                except Exception as e:

                    st.error(f"❌ Error: {e}")

elif tab == "🛠️ Code Debugger":

    if st.button("🛠️ Debug Code"):

        if not code\_input.strip():

            st.warning("Please paste some code first.")

        else:

            with st.spinner("🕵️‍♂️ Debugging your code..."):

                prompt = f"Find and explain any possible errors in the following {language} code. Suggest corrections:\n\n{code\_input}"

                try:

                    response = model.generate\_content(prompt)

                    debug\_output = response.text

                    st.markdown("### 🧪 Debugging Report:")

                    st.error(debug\_output)

                    st.markdown("### 📄 Your Code:")

                    st.code(code\_input, language=language.lower())

                    st.markdown(get\_download\_link(debug\_output), unsafe\_allow\_html=True)

                    st.snow()

                except Exception as e:

                    st.error(f"❌ Error: {e}")

st.markdown("<hr><center>Made with ❤️ by Krishna Gupta · Powered by Gemini 1.5 · Deployed on Streamlit</center>", unsafe\_allow\_html=True)

APP AND OUTPUT SCREENSHOT:











