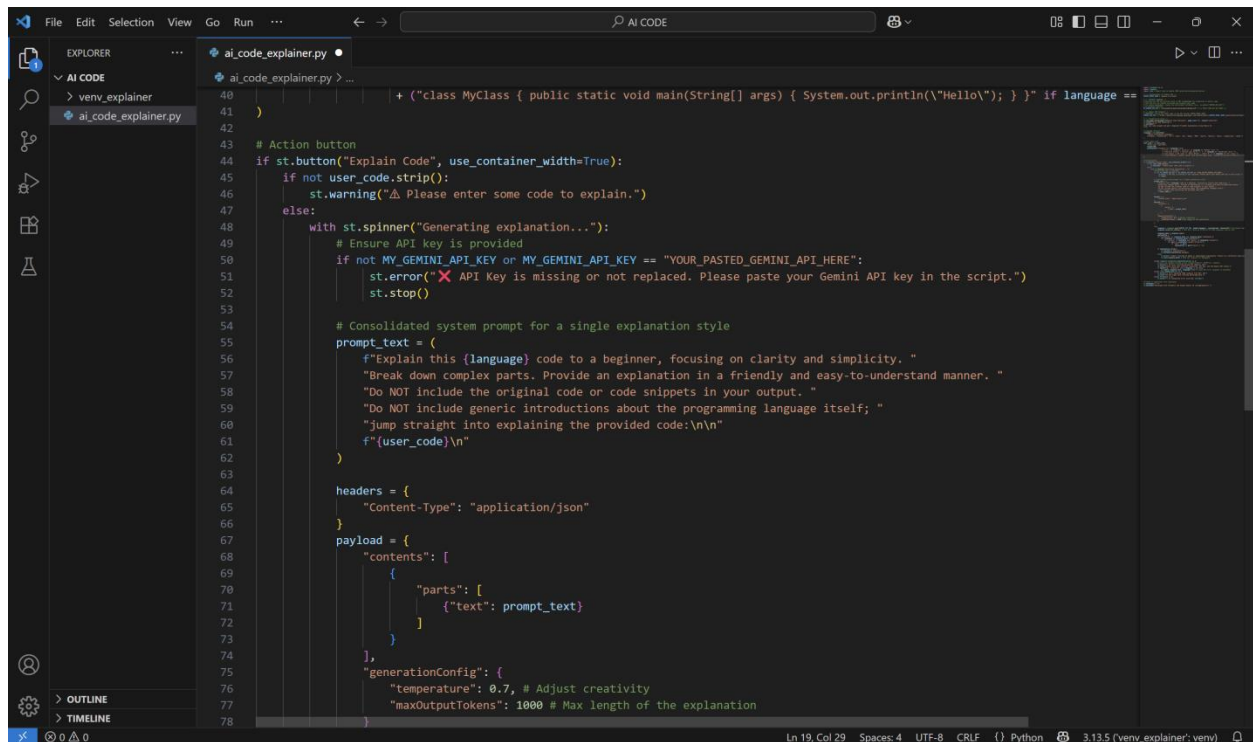


Name :- B.Tejaswini

Clg Name :- VIT (vellore)

## AI Code Explainer

Code :-



```
40 + ("class MyClass { public static void main(String[] args) { System.out.println(\"Hello\"); } }" if language ==
41 )
42
43 # Action button
44 if st.button("Explain Code", use_container_width=True):
45     if not user_code.strip():
46         st.warning("⚠ Please enter some code to explain.")
47     else:
48         with st.spinner("Generating explanation..."):
49             # Ensure API key is provided
50             if not MY_GEMINI_API_KEY or MY_GEMINI_API_KEY == "YOUR_PASTED_GEMINI_API_HERE":
51                 st.error("❌ API Key is missing or not replaced. Please paste your Gemini API key in the script.")
52                 st.stop()
53
54             # Consolidated system prompt for a single explanation style
55             prompt_text = (
56                 f"Explain this {language} code to a beginner, focusing on clarity and simplicity. "
57                 "Break down complex parts. Provide an explanation in a friendly and easy-to-understand manner. "
58                 "Do NOT include the original code or code snippets in your output. "
59                 "Do NOT include generic introductions about the programming language itself; "
60                 "jump straight into explaining the provided code:\n\n"
61                 f"{user_code}\n"
62             )
63
64             headers = {
65                 "Content-Type": "application/json"
66             }
67             payload = {
68                 "contents": [
69                     {
70                         "parts": [
71                             {"text": prompt_text}
72                         ]
73                     },
74                 ],
75                 "generationConfig": {
76                     "temperature": 0.7, # Adjust creativity
77                     "maxOutputTokens": 1000 # Max length of the explanation
78                 }
79             }
```


```
File Edit Selection View Go Run ... AI CODE
EXPLORER
  AI CODE
    venv_explainer
      ai_code_explainer.py
ai_code_explainer.py
1 import streamlit as st
2 import requests
3 import json # Import json to handle JSON serialization/deserialization
4
5 # --- Configuration for Gemini API ---
6 GEMINI_MODEL_NAME = "gemini-1.5-flash"
7
8 # !!! SECURITY WARNING !!!
9 # Pasting your API key directly here is NOT recommended for production or public code.
10 # This key will be visible to anyone who accesses your code.
11 # For secure deployment, always use environment variables (e.g., os.getenv("GEMINI_API_KEY"))
12 # or Streamlit Secrets.
13 MY_GEMINI_API_KEY = "AIZA5yAdGveG_WqxPsudYuZbLbLIIMZnUmltjM" # <<< PASTE YOUR API KEY HERE >>>
14
15 # --- Gemini API Endpoint ---
16 # IMPORTANT: Ensure the model name in the URL matches GEMINI_MODEL_NAME
17 GEMINI_API_URL = f"https://generativelanguage.googleapis.com/v1beta/models/{GEMINI_MODEL_NAME}:generateContent?key={MY_GEMINI_API_KEY}"
18
19 # --- Streamlit UI setup ---
20 st.set_page_config(page_title="AI Code Explainer", page_icon="🤖", layout="centered")
21 st.title("🤖 AI Code Explainer")
22 st.markdown("""
23 Enter any code snippet and get a beginner-friendly explanation using Gemini AI.
24 """)
25
26 # Language selector
27 language = st.selectbox(
28     "Choose Programming Language:",
29     ["Python", "JavaScript", "C++", "Java", "Go", "Ruby", "PHP", "Swift", "Kotlin", "Rust", "TypeScript", "Other"]
30 )
31
32 # Code input area
33 user_code = st.text_area(
34     "Paste your code here",
35     height=300,
36     placeholder=f"Example for {language}:\n\n"
37     + ("for i in range(5): print(i) if language == 'Python' else """)
38     + ("function greet() { console.log('Hello!'); } if language == 'JavaScript' else """)
39     + ("int main() { std::cout << 'Hello World!'; return 0; } if language == 'C++' else """)

```

```
79
80
81 try:
82     response = requests.post(GEMINI_API_URL, headers=headers, json=payload, timeout=60) # Increased timeout
83     response.raise_for_status() # This will raise an HTTPError for bad responses (4xx or 5xx)
84
85     response_data = response.json()
86     explanation = ""
87     if "candidates" in response_data and response_data["candidates"]:
88         for candidate in response_data["candidates"]:
89             if "content" in candidate and "parts" in candidate["content"]:
90                 for part in candidate["content"]["parts"]:
91                     if "text" in part:
92                         explanation += part["text"] + "\n"
93
94     if explanation.strip():
95         st.success("✅ Explanation:")
96         st.markdown(explanation.strip())
97     else:
98         st.error("❌ Gemini returned an empty or unparseable explanation. Please try a different code snippet or prompt.")
99         st.json(response_data) # Show raw response for debugging
100
101 except requests.exceptions.RequestException as e:
102     # Catch any request-related errors (ConnectionError, HTTPError, Timeout)
103     st.error(f"❌ An error occurred during API request: {e}")
104     st.info("Please check your internet connection, API key, and the Gemini API status.")
105     if hasattr(e, 'response') and e.response is not None:
106         st.code(e.response.text, language="json") # Show API error response if available
107 except json.JSONDecodeError as e:
108     st.error(f"❌ Error decoding JSON response from API: {e}")
109     st.info("The API might have returned malformed data.")
110 except Exception as e:
111     st.error(f"❌ An unexpected error occurred: {str(e)}")
112
113 # Footer or additional info (optional)
114 st.markdown("----")
115 st.markdown("Developed with Streamlit and Google Gemini AI (using requests).")
116
```

Output:-

Deploy

 **AI Code Explainer**

Enter any code snippet and get a beginner-friendly explanation using Gemini AI.

Choose Programming Language:  
Python


Paste your code here

Example for Python:  

```
for i in range(5): print(i)
```

Explain Code

Deploy

 **AI Code Explainer**

Enter any code snippet and get a beginner-friendly explanation using Gemini AI.

Choose Programming Language:  
Python

Paste your code here

```
n = int(input("Enter an integer: "))  
if n % 2 == 0:  
    print("Even")  
else:  
    print("Odd")
```

Explain Code

### Explain Code

#### ✓ Explanation:

This program checks if a number is even or odd.

First, it asks the user to enter a whole number (an integer). The program then stores that number in a variable called 'n'.

Next, it uses the modulo operator (`%`). This operator gives you the remainder after a division. For example, `10 % 3` is 1 (because 10 divided by 3 is 3 with a remainder of 1).

The program checks if the remainder of 'n' divided by 2 is equal to 0. If it is, that means the number is perfectly divisible by 2, and therefore even. In this case, the program prints "Even".

If the remainder is not 0 (meaning it's 1), then the number is odd, and the program prints "Odd". It's a simple way to determine even or odd numbers using a fundamental math concept.

---

Developed with Streamlit and Google Gemini AI (using requests).



## AI Code Explainer

Enter any code snippet and get a beginner-friendly explanation using Gemini AI.

Choose Programming Language:

JavaScript

Paste your code here

```
function greet(name) {  
  return "Hello, " + name + "!";  
}  
  
console.log(greet("World"));  
console.log(greet("Alice"));
```

Explain Code

✓ Explanation:

Imagine you have a little machine that takes a name as input and says hello to that name. That's what the code does!

It defines a "recipe" (called a function) named `greet`. This recipe has one step: it takes whatever name you give it and creates a greeting like "Hello, [name]!". For example, if you give it "World", it makes "Hello, World!".

Then, the code uses this recipe twice. First, it feeds the name "World" to the `greet` recipe, and the resulting "Hello, World!" is shown on the screen. Second, it uses the name "Alice", resulting in "Hello, Alice!" being displayed on the screen. It's like using a stamp to make two different greeting cards.

---

Developed with Streamlit and Google Gemini AI (using requests).

✓ Explanation:

Imagine you have a little machine that takes a name as input and says hello to that name. That's what the code does!

It defines a "recipe" (called a function) named `greet`. This recipe has one step: it takes whatever name you give it and creates a greeting like "Hello, [name]!". For example, if you give it "World", it makes "Hello, World!".

Then, the code uses this recipe twice. First, it feeds the name "World" to the `greet` recipe, and the resulting "Hello, World!" is shown on the screen. Second, it uses the name "Alice", resulting in "Hello, Alice!" being displayed on the screen. It's like using a stamp to make two different greeting cards.

---

Developed with Streamlit and Google Gemini AI (using requests).