

Code pilot



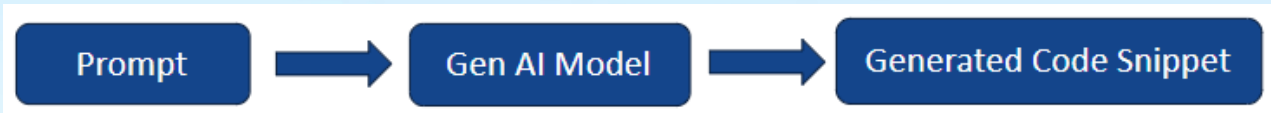
Content

- Introduction
- User Interface Details – CodePilot
- Flow Diagram of Generative AI for text generation
- Why to Use CodePilot
- Features of CodePilot
- Benefits and applications of CodePilot
- Procedure to Use CodePilot
- Check Codepilot output in IDE

Introduction

- **Welcome** to the CodePilot Application! This application is designed to assist developers by providing intelligent code completion as they write code. This user manual will guide you through its usage and customization.
- In today's fast-changing tech world, AI-powered solutions are like super helpers for changing industries.
- One shining example is CodePilot, which uses fancy technology called Open Source Large Language Models (Generative AI).
- It is a smart mix of tech that wants to make coding easier, faster, and more creative.
- This smarts tool is here to make tech stuff more friendly and exciting for all of us!

User Interface Details - CodePilot



Test Cases

The screenshot shows the CodePilot interface. On the left, there is a 'Test Cases' section with a toggle switch and a dropdown menu containing 'Write python function for palindrome number'. Below this is a 'GENERATE CODE' button. Underneath, there is a prompt area with the text 'Write your prompt to generate code' and 'Write your prompt here', followed by a text input field containing 'write program for fibonacci series using java' and another 'GENERATE CODE' button. A red arrow points from the 'Test Cases' label to the dropdown menu. Another red arrow points from the 'Custom Test Cases' label to the text input field. On the right, the 'AI Based CodePilot' section displays a code snippet for a Java Fibonacci program. A red arrow points from the 'CodePilot Output-Generated Code' label to the code snippet.

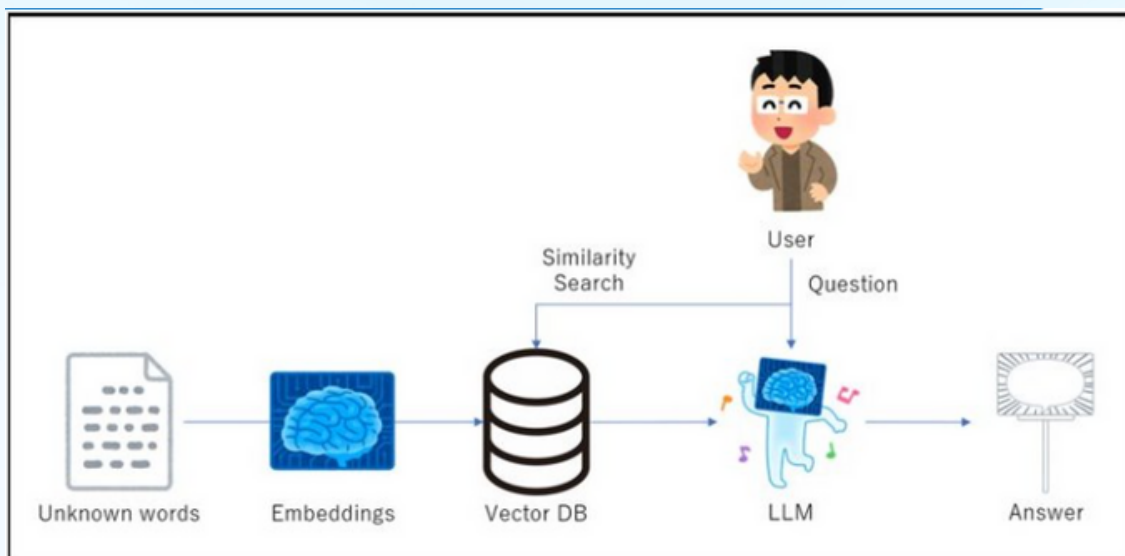
```
Here is a Java program that calculates the fibonacci series:  
...  
public class Fibonacci {  
    public static void main(String[] args) {  
        int n = Integer.parseInt(args[0]);  
        System.out.println("The " + n + "th Fibonacci number is " + fibonacci(  
    })  
}
```

CodePilot Output-Generated Code

Custom Test Cases

Flow diagram of Generative AI for text generation

Generative AI operates by first training on a dataset, learning patterns and relationships within it. After training, the AI model can generate content based on input data. This process includes data preprocessing, model selection, training, fine-tuning, and evaluation. Once deployed, the AI requires ongoing monitoring and ethical considerations to ensure responsible use and continuous improvement.



Reference-https://qiita.com/t_serizawa/items/6613ea0bc2a13822f5cb

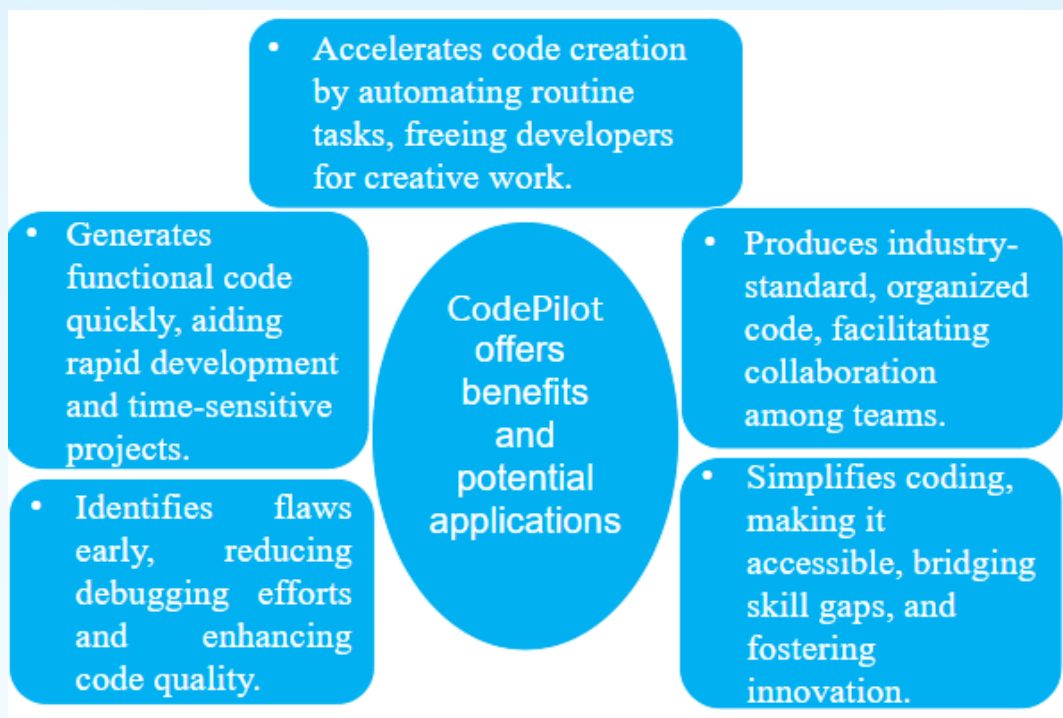
Why to Use CodePilot?

- CodePilot is a tool powered by artificial intelligence designed to aid engineers in writing code more efficiently and accurately.
- It generates code in real-time based on prompts or provides suggestions for code auto-completion.
- This is capable of translating code between different programming languages and creating programs.
- CodePilot utilizes algorithms trained on open-source project code examples to create new code based on those examples.
- Select optimal code for specific programs, accelerating the development process for programmers.
- CodePilot operates through auto-completion, suggestion algorithms, and direct communication between developers and AI.

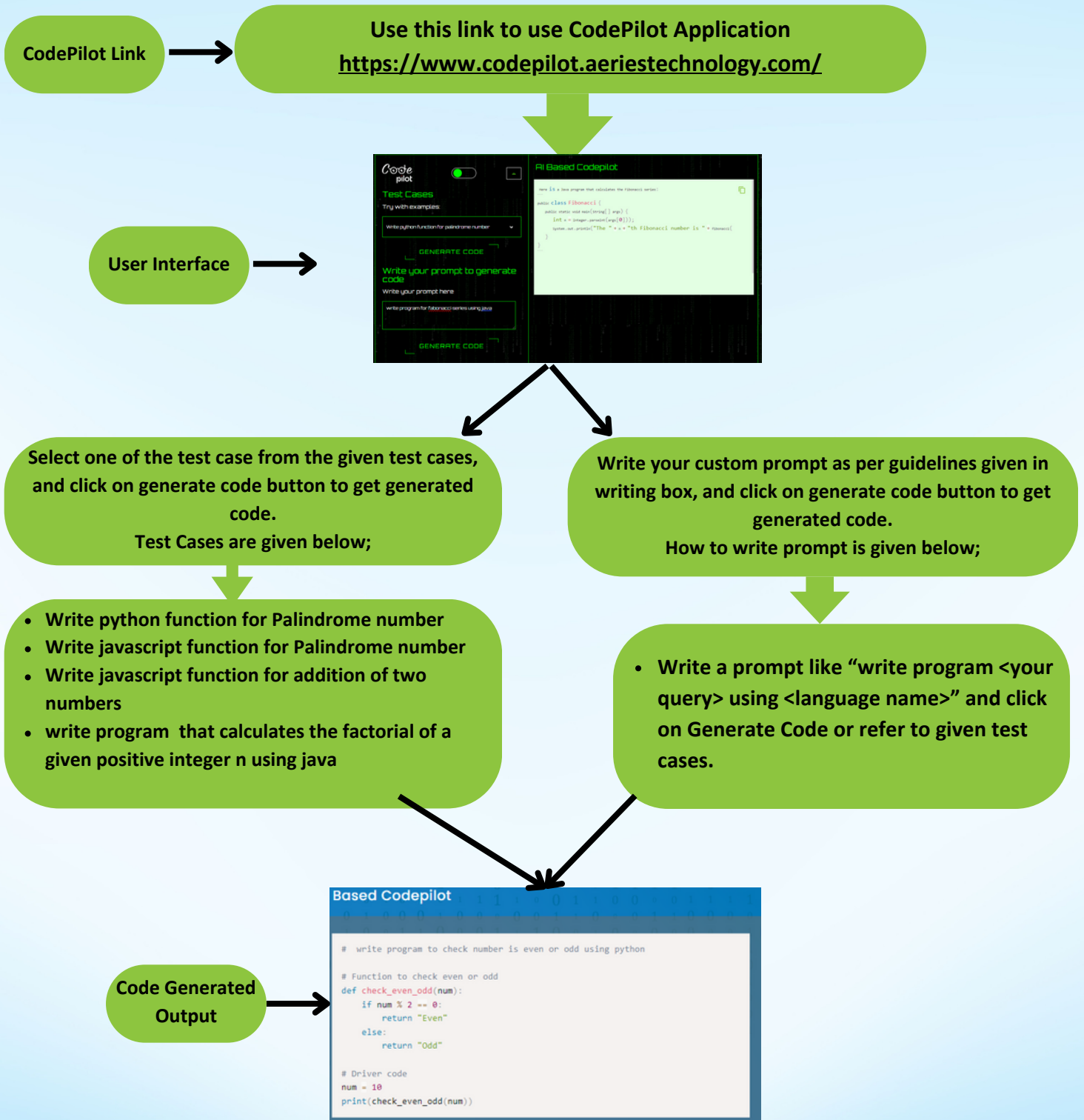
Features of CodePilot

- Making comments and alerts while coding creates secure and dependable codes.
- It reduces the repetitious nature of coding.
- It can generate code for machine learning models, data science, and other applications.
- It supports different programming languages viz python, java, javascript, c, c++, Node.js etc.
- It helps in application development.
- It can cater to many developers as it supports different languages.
- Generating code is more straightforward, with an intuitive methodology and user-friendly UI.

Benefits and applications of CodePilot



Procedure to Use CodePilot



Check CodePilot output in IDE

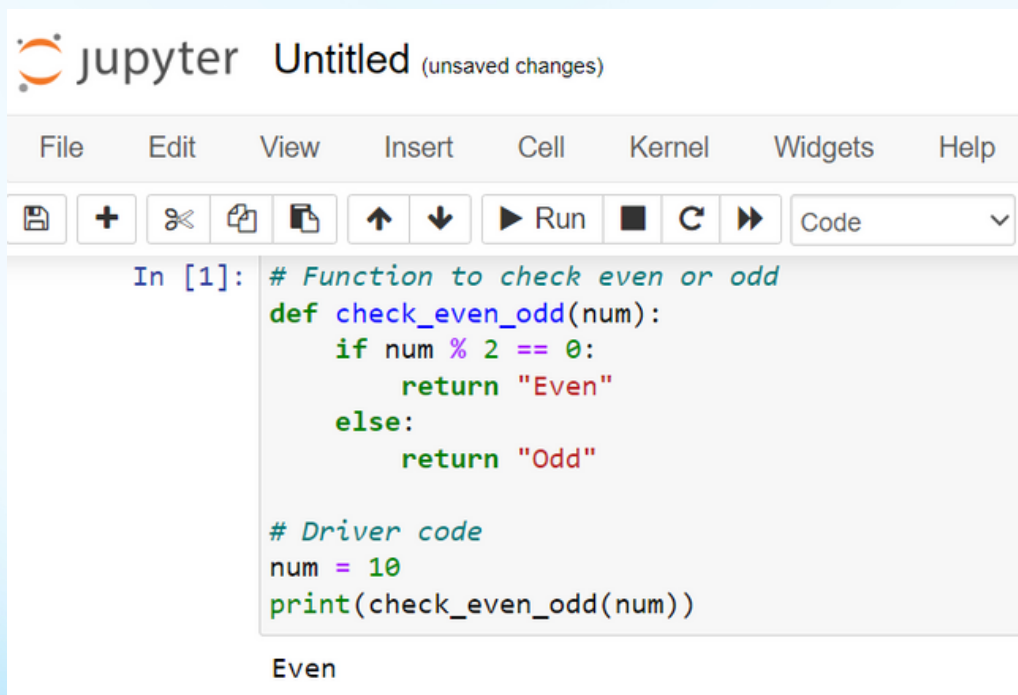
CodePilot Output

```
# write program to check number is even or odd using python

# Function to check even or odd
def check_even_odd(num):
    if num % 2 == 0:
        return "Even"
    else:
        return "Odd"

# Driver code
num = 10
print(check_even_odd(num))
```

Any IDE Output



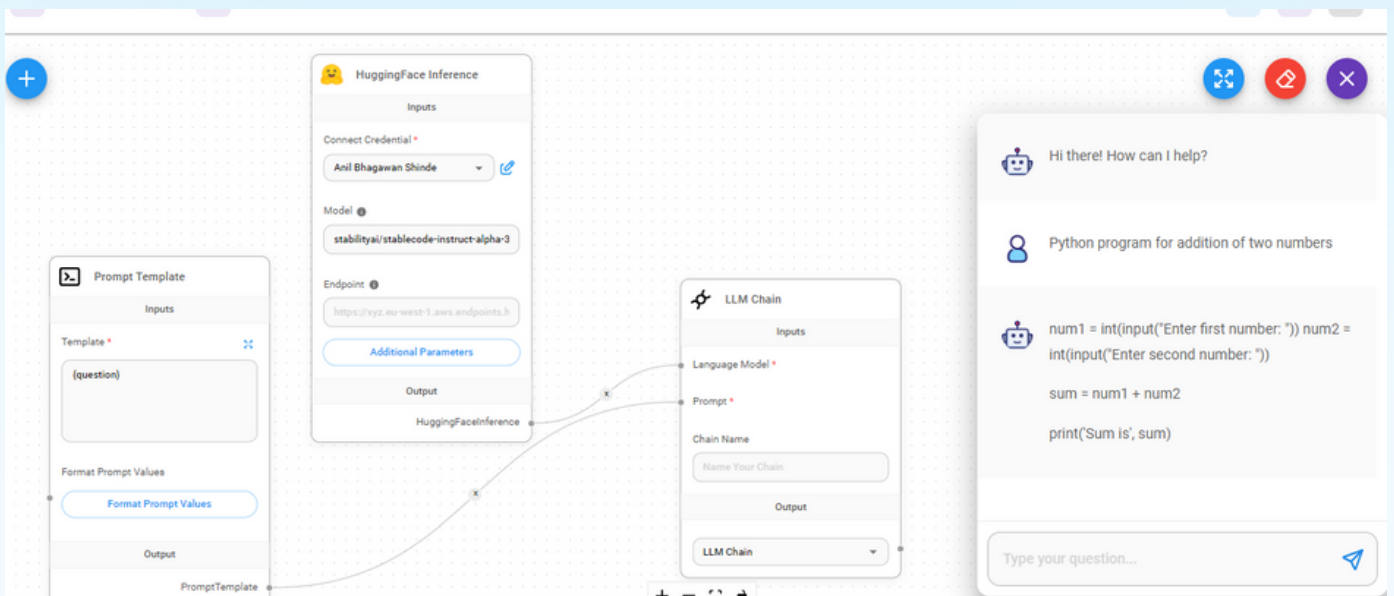
The screenshot shows a Jupyter IDE window titled "Untitled (unsaved changes)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The toolbar contains icons for file operations, a Run button, and a dropdown menu set to "Code". The code cell contains the same Python code as in the previous block. Below the code, the output "Even" is displayed.

```
In [1]: # Function to check even or odd
def check_even_odd(num):
    if num % 2 == 0:
        return "Even"
    else:
        return "Odd"

# Driver code
num = 10
print(check_even_odd(num))

Even
```

CodePilot Using No-Code Platform



<https://buildergpt.aeriestechnology.com/canvas/6daf5108-d68e-4e0d-884b-12945eb68c79>