

KIET GROUP OF INSTITUTIONS



ACADEMIC SESSION 2024-2025 INTRODUCTION TO AI PRIME NUMBER GENERATOR AND CHECKER WITH VISUALIZATION

NAME	Ayush Rao Chaudhary
BRANCH	CSE (AI & ML)
SEC	A
ROLL NO	67
SUBJECT	Introduction To AI
SUB CODE	AI101B
ASSIGNMENT GUIDE	MR Bikki Gupta Sir

INTRODUCTION

Prime numbers play a crucial role in various fields of mathematics, cryptography, and computer science. A prime number is a natural number greater than 1 that has no divisors other than 1 and itself.

In this project, we develop a Prime Number Generator and Checker using Python. The implementation follows two key functionalities:

- 1. Prime Checker - Determines whether a given number is prime.*
- 2. Prime Generator - Generates all prime numbers within a given range.*

Additionally, we integrate data visualization using Matplotlib and Pandas to provide an insightful representation of prime numbers.

PROBLEM STATEMENT

Given a number or a range of numbers, we need to:

- ✓ Check if a specific number is prime.
- ✓ Generate all prime numbers in a specified range.
- ✓ Visualize the results using graphs and plots.

METHODOLOGY

To accomplish this task, we follow a structured approach:

Step 1: Implement a Prime Checker Function

- Define a function `is_prime(n)` that checks if n is prime.
 - Use a loop from 2 to $\text{sqrt}(n)$ to check divisibility.
 - If n is divisible by any number in this range, return `False`, else return `True`.

Step 2: Implement a Prime Generator Function

- Define a function `generate_primes(start, end)` to find primes in the range $[start, end]$.
 - Use `is_prime(n)` to check for prime numbers.
 - Store the prime numbers in a list.

Step 3: Store the Results Using Pandas

- Store the generated prime numbers in a Pandas DataFrame.
 - Assign index values to track prime numbers in sequential order.

Step 4: Visualize Using Matplotlib

- Use Bar Chart to represent prime numbers.
- Use Scatter Plot to show individual prime numbers.
- Use Line Plot to connect them for better pattern recognition.

Step 5: Display the Results

- Print the list of prime numbers.
- Generate a graph for visual representation.

CODE

IMPLEMENTATION

**# Prime Number Generator and Checker with
Visualization**

```
import matplotlib.pyplot as plt  
import pandas as pd
```

Function to check if a number is prime

```
def is_prime(n):  
  
    if n < 2:  
  
        return False # Numbers less than 2 are not  
prime  
  
    for i in range(2, int(n**0.5) + 1):  
  
        if n % i == 0:  
  
            return False # If divisible, not a prime  
number  
  
    return True # If no divisors found, it is prime
```

```
# Function to generate prime numbers in a given range
def generate_primes(start, end):
    primes = [] # Initialize an empty list to store prime
    numbers
    for num in range(start, end + 1):
        if is_prime(num): # Check if the number is prime
            primes.append(num) # Add it to the list
    return primes # Return the list of prime numbers
```

```
# Example usage
```

```
start=int(input("Enter Lower Limit : "))
end = int(input("Enter Upper Limit : ")) # Define range
for prime number generation
prime_numbers = generate_primes(start, end)
print("Prime Numbers In The Given Range Are :
",end="")
print(prime_numbers)
```

```
# Checking if a specific number is prime
```

```
num = int(input("Enter A Number To Check It Is Prime  
Or Not : "))
```

```
# Define a number to check
```

```
print(f"Is {num} a prime number? {is_prime(num)}")
```

```
# Output whether the number is prime or not
```

```
# Create a DataFrame for visualization
```

```
df = pd.DataFrame({'Prime Numbers': prime_numbers,  
'Index': range(1, len(prime_numbers) + 1)})
```

```
# Plot the prime numbers
```

```
plt.figure(figsize=(10, 6))
```

```
bar_plot = plt.bar(df['Index'], df['Prime Numbers'],  
color='purple', alpha=0.7, label='Prime Numbers')
```

```
scatter_plot = plt.scatter(df['Index'], df['Prime  
Numbers'], color='red', label='Prime Points')
```

```
line_plot, = plt.plot(df['Index'], df['Prime Numbers'],  
linestyle='dashed', color='blue', alpha=0.5, label='Prime  
Trend')
```

```
# Customizing the graph
```

```
plt.xlabel('Index')
```

```
plt.ylabel('Prime Numbers')
```

```
plt.title('Visualization of Prime Numbers')
```

```
plt.legend(handles=[bar_plot, scatter_plot, line_plot])
```

```
plt.grid(True, linestyle='--', alpha=0.6)
```

```
plt.show()
```

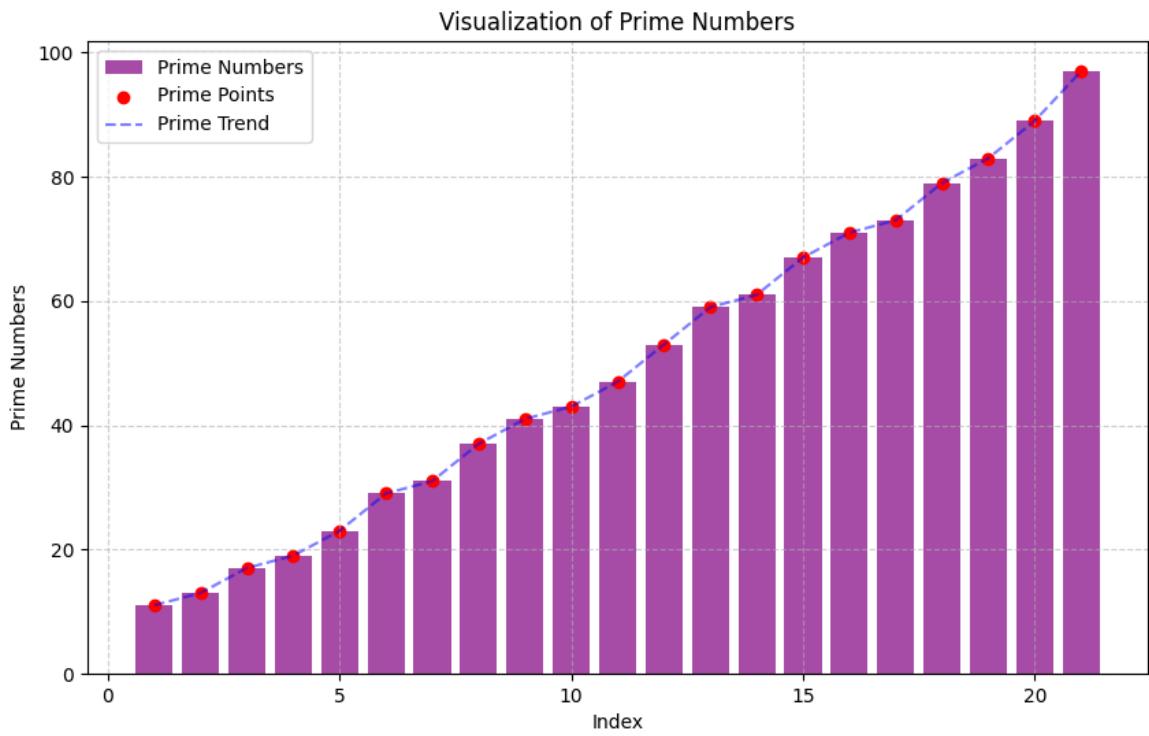
OUTPUT & RESULTS

Prime Numbers in the Range (10-100)

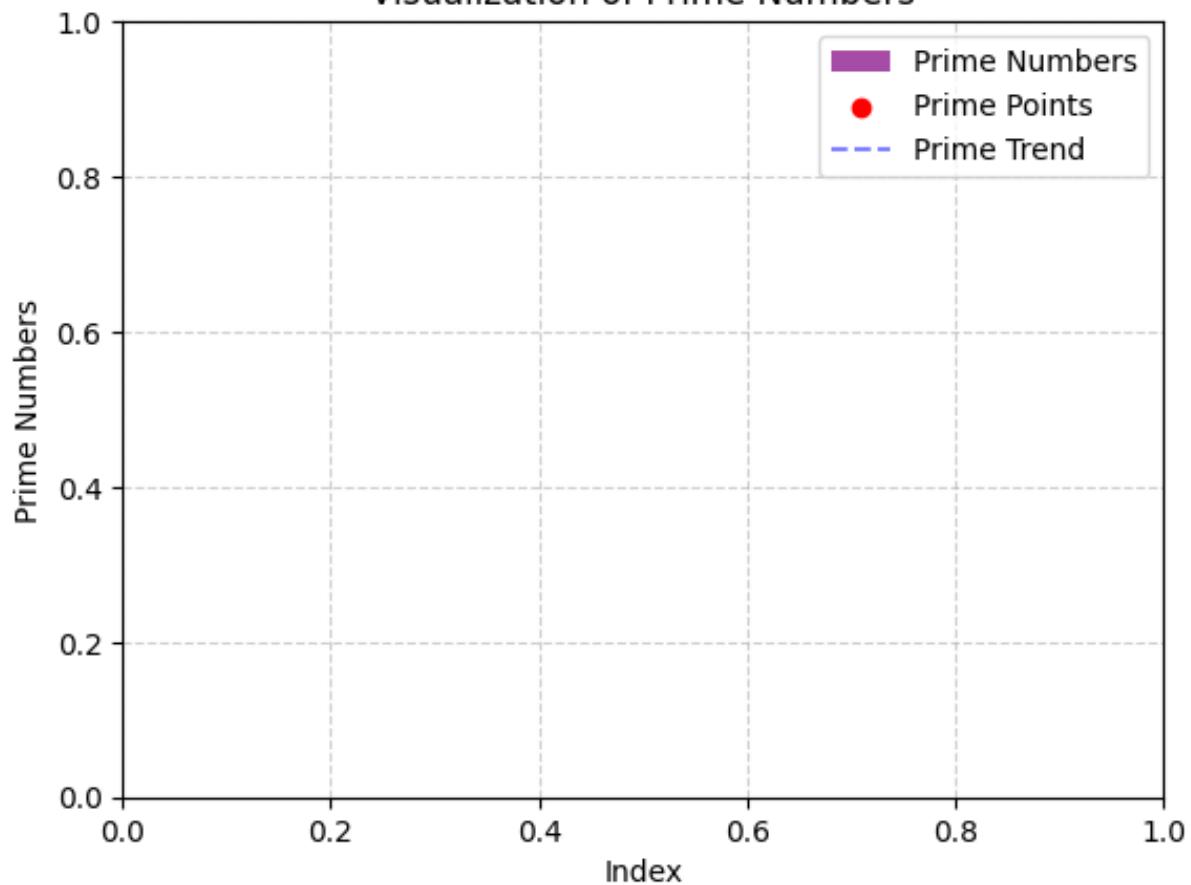
```
PS X:\VS CODE> python -u "x:\VS CODE\TEMPO.PY"
Enter Lower Limit : 10
Enter Upper Limit : 100
Prime Numbers In The Given Range Are : [11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89,
97]
Enter A Number To Check It Is Prime Or Not : 33
Is 33 a prime number? False
PS X:\VS CODE>
```

Ln 36, Col 3 Spaces: 4 UTF-8 CRLF {} Python 3.11.9 64-bit (Microsoft Store) ⚙ Prettier

Graph Representation



Visualization of Prime Numbers



REFERENCES & CREDITS

◦ Python Documentation:

<https://docs.python.org/3/>

◦ Matplotlib Documentation:

<https://matplotlib.org/>

◦ Pandas Documentation:

<https://pandas.pydata.org/>

◦ Number Theory Concepts:

https://en.wikipedia.org/wiki/Prime_number