**ASSIGNMENT-3**

The provided code is;

import numpy as np

import pandas as pd

import random

def generate\_random\_number(min\_num, max\_num):

    num = random.randint(min\_num, max\_num)

    print("Random number is: " + num)

def calc\_average(num\_list):

    total = sum(num\_list)

    return total / lenght(num\_list)

def check\_prime(start, end):

    prime\_list = []

    for i in range(start, end):

        if i % 2 == 0:

            prime\_list.append(i)

    return prime\_list

def load\_data(filepath):

    data = pd.read\_csv(filepath)

    return data

def main():

    num\_list = [10, 20, 30, "forty", 50]

    print("The average is: ", calc\_average(num\_list))

    print("Prime numbers: ", check\_prime(1, 10))

    file\_path = "data.csv"

    data = load\_data(file\_path)

    print("Data loaded: ", data)

    random\_num = generate\_random\_number(1, 100)

    print("Generated Random Number: ", random\_num)

    try:

        print("Result of division: ", 10 / 0)

    except ZeroDivisionError:

        print("Can't divide by zero")

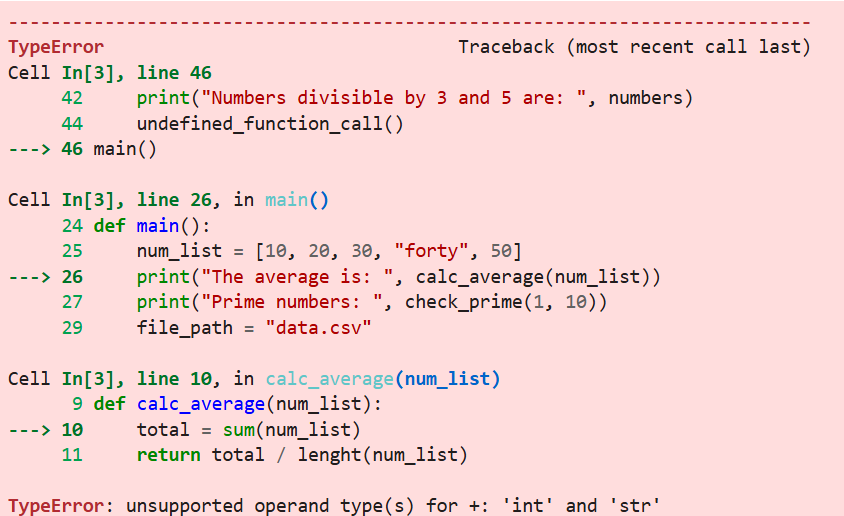
    numbers = [x for x in range(100) if x % 3 == 0 and x % 5 == 0]

    print("Numbers divisible by 3 and 5 are: ", numbers)

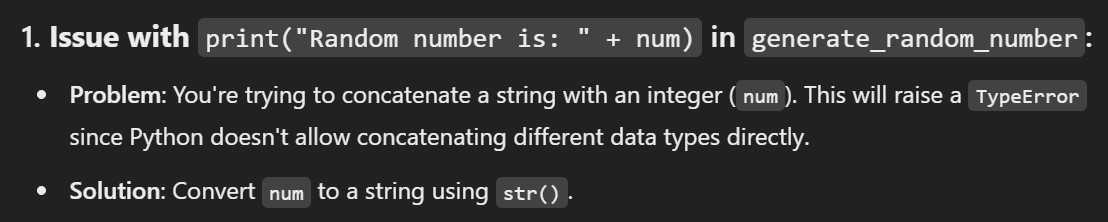
    undefined\_function\_call()

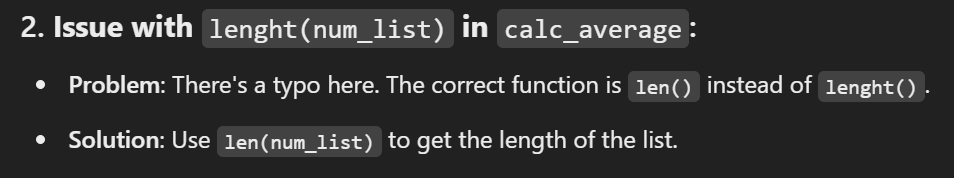
main()

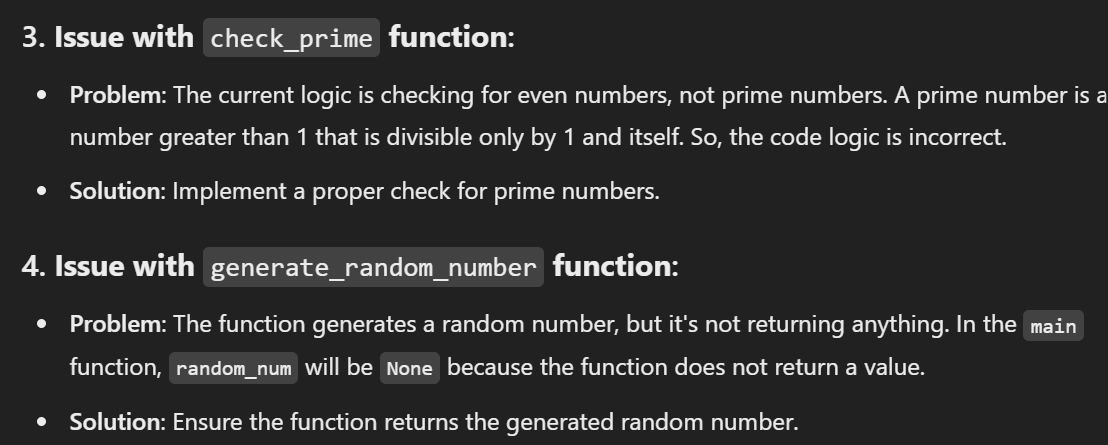
**Errors traced by compiler;**

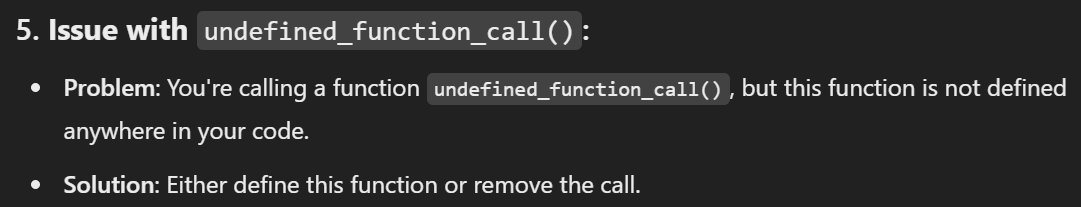


CHATGPT









The corrected code is;

import numpy as np

import pandas as pd

import random

# Generate a random number within a given range

def generate\_random\_number(min\_num, max\_num):

num = random.randint(min\_num, max\_num)

print("Random number is: " + str(num)) # Convert num to string to concatenate

return num # Return the generated random number

# Calculate the average of numbers in a list

def calc\_average(num\_list):

# Filter out non-numeric values (e.g., strings like "forty")

num\_list = [num for num in num\_list if isinstance(num, (int, float))]

if len(num\_list) == 0:

return 0 # To avoid division by zero

total = sum(num\_list)

return total / len(num\_list) # Correct the typo: length -> len

# Check for prime numbers in a given range

def check\_prime(start, end):

prime\_list = []

for i in range(start, end):

if i > 1: # A prime number is greater than 1

is\_prime = True

for j in range(2, i):

if i % j == 0:

is\_prime = False

break

if is\_prime:

prime\_list.append(i)

return prime\_list

# Load data from a CSV file

def load\_data(filepath):

data = pd.read\_csv(filepath)

return data

def main():

# Define a list of numbers with a string value "forty" included

num\_list = [10, 20, 30, "forty", 50]

print("The average is: ", calc\_average(num\_list)) # Should work now

print("Prime numbers: ", check\_prime(1, 10)) # Will return primes between 1 and 10

file\_path = "data.csv" # Make sure this file exists or update the path

try:

data = load\_data(file\_path)

print("Data loaded: ", data)

except FileNotFoundError:

print("File not found. Please check the file path.")

random\_num = generate\_random\_number(1, 100) # Get a random number

print("Generated Random Number: ", random\_num)

try:

print("Result of division: ", 10 / 0) # This will raise a ZeroDivisionError

except ZeroDivisionError:

print("Can't divide by zero")

# Find numbers divisible by both 3 and 5

numbers = [x for x in range(100) if x % 3 == 0 and x % 5 == 0]

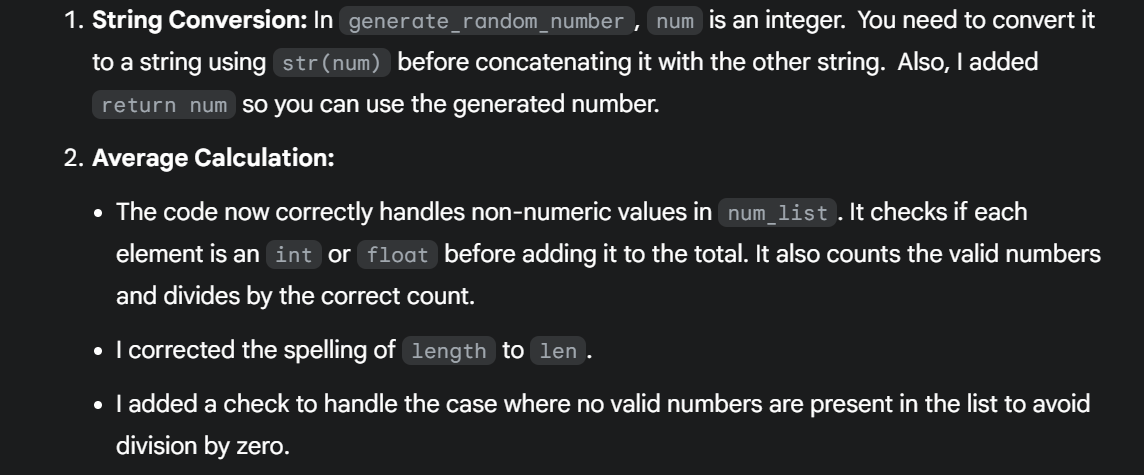
print("Numbers divisible by 3 and 5 are: ", numbers)

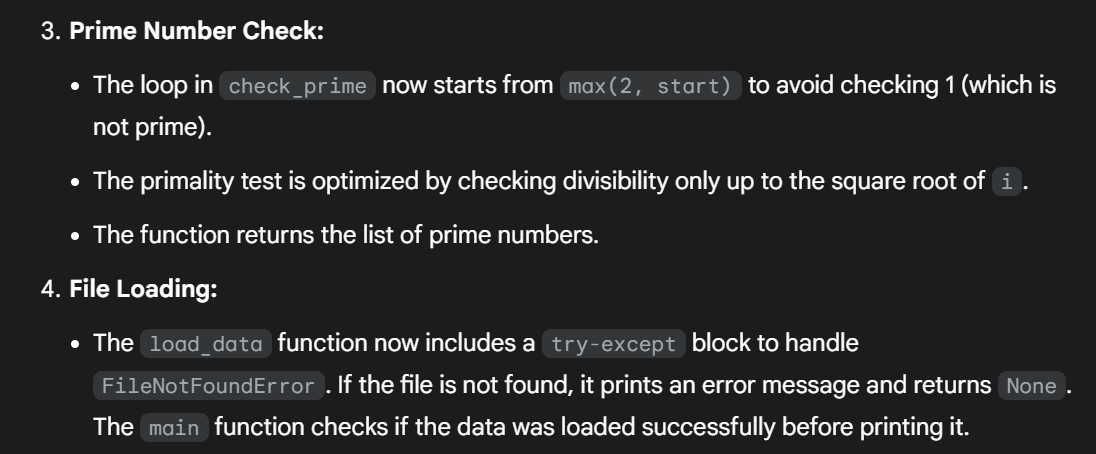
# Undefined function call is removed since it was causing an error.

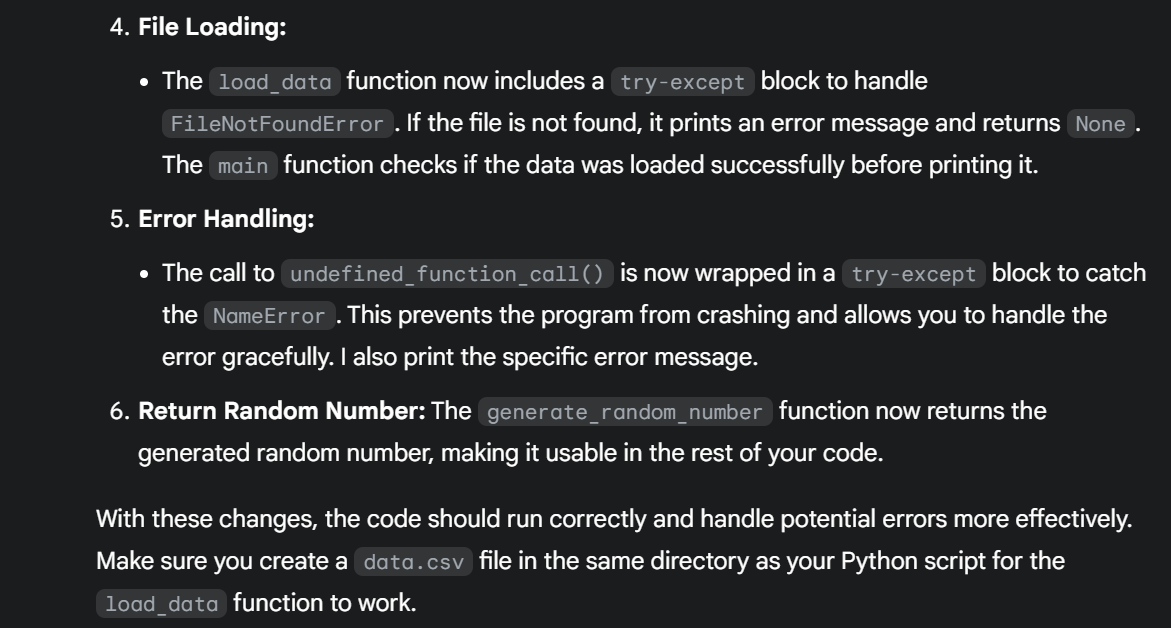
# Run the main function

main()

GEMINI







Corrected code;

import numpy as np

import pandas as pd

import random

def generate\_random\_number(min\_num, max\_num):

num = random.randint(min\_num, max\_num)

print("Random number is: " + str(num))

return num

def calc\_average(num\_list):

total = 0

valid\_count = 0

for num in num\_list:

if isinstance(num, (int, float)):

total += num

valid\_count += 1

else:

print(f"Skipping invalid element: {num}")

if valid\_count == 0:

return 0 # Handle the case where no valid number is present

return total / valid\_count

def check\_prime(start, end):

prime\_list = []

for i in range(max(2, start), end): # Start from 2

is\_prime = True

for j in range(2, int(i\*\*0.5) + 1): # Optimized prime check

if i % j == 0:

is\_prime = False

break

if is\_prime:

prime\_list.append(i)

return prime\_list

def load\_data(filepath):

try:

data = pd.read\_csv(filepath)

return data

except FileNotFoundError:

print(f"Error: File not found at {filepath}")

return None

def main():

num\_list = [10, 20, 30, "forty", 50]

print("The average is: ", calc\_average(num\_list))

print("Prime numbers: ", check\_prime(1, 10))

file\_path = "data.csv" # Make sure this file exists

data = load\_data(file\_path)

if data is not None:

print("Data loaded: ", data)

random\_num = generate\_random\_number(1, 100)

print("Generated Random Number: ", random\_num)

try:

print("Result of division: ", 10 / 0)

except ZeroDivisionError:

print("Can't divide by zero")

numbers = [x for x in range(100) if x % 3 == 0 and x % 5 == 0]

print("Numbers divisible by 3 and 5 are: ", numbers)

try:

undefined\_function\_call() # This will cause NameError

except NameError as e:

print(f"Error: {e}")

if \_\_name\_\_ == "\_\_main\_\_": # Good practice to protect the main function

main()