## CA 2 Generative Ai

Name: Sanjula Kalbande

PRN: 21070521070

Section: B

Q:2 Generate a model in Python to represent a Housing loan scheme and create a chart to

display the Emi based on rate of interest and reducing balance for a given period. If a customer

wishes to close the loan earlier, print the interest lost distributed over the remaining no. Of

months. Assume suitable data and inputs as necessary. import matplotlib.pyplot as plt

def calculate\_emi(principal, rate, tenure):

  r = rate / 12 / 100

  n = tenure \* 12

  emi = principal \* r \* (1 + r) \*\* n / ((1 + r) \*\* n - 1)

  return emi

def calculate\_interest\_lost(principal, rate, remaining\_tenure):

  remaining\_emi = calculate\_emi(principal, rate, remaining\_tenure)

  total\_interest\_paid = remaining\_emi \* remaining\_tenure \* 12 - principal

  return total\_interest\_paid

def display\_emi\_chart(principal, rate, tenure):

emi = calculate\_emi(principal, rate, tenure)

months = range(1, tenure \* 12 + 1)

interest\_paid = []

principal\_paid = []

balance = principal

for month in months:

interest = balance \* rate / 12 / 100

principal\_payment = emi - interest

balance -= principal\_payment

interest\_paid.append(interest)

principal\_paid.append(principal\_payment)

plt.plot(months, interest\_paid, label="Interest Paid")

plt.plot(months, principal\_paid, label="Principal Paid")

plt.xlabel("Months")

plt.ylabel("Amount")

plt.title("EMI Breakdown")

plt.legend()

plt.show()

principal = float(input("Enter the principal amount: "))

rate = float(input("Enter the annual interest rate (%): "))

tenure = int(input("Enter the loan tenure (years): "))

emi = calculate\_emi(principal, rate, tenure)

print("EMI:", emi)

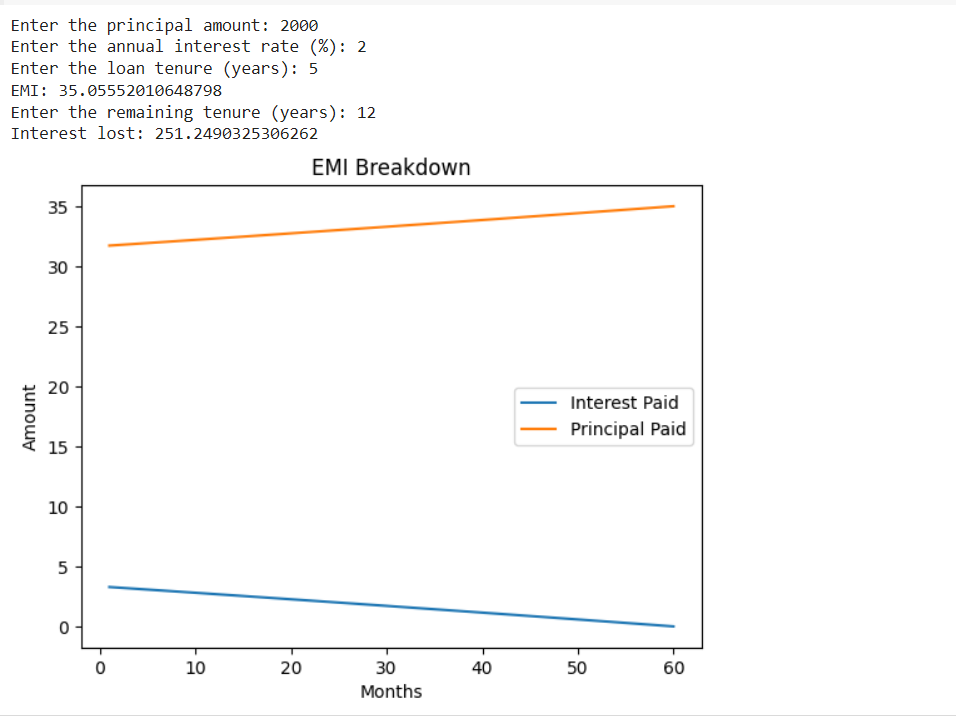
remaining\_tenure = int(input("Enter the remaining tenure (years): "))

interest\_lost = calculate\_interest\_lost(principal, rate, remaining\_tenure)

print("Interest lost:", interest\_lost)

display\_emi\_chart(principal, rate, tenure)

OUTPUT:



Q:3 Generate a model for an Insurance company to hold information on the insurer's vehicle,

and create a chart of monthly, yearly, and qtrly premiums based on no. of years of insurance

where in each year, the value of the vehicle depreciates by 7%.

import matplotlib.pyplot as plt

def calculate\_premium(vehicle\_type, year, coverage\_type, term):

base\_premium = 1000

if vehicle\_type == "car":

base\_premium += 500

if year > 5:

base\_premium -= 100 \* (year - 5)

if coverage\_type == "comprehensive":

base\_premium \*= 1.5

if term == "monthly":

base\_premium \*= 1.1

elif term == "quarterly":

base\_premium \*= 1.05

return base\_premium

def generate\_premium\_chart(vehicle\_type, year, coverage\_type, term, num\_years):

premiums = []

for \_ in range(num\_years):

premium = calculate\_premium(vehicle\_type, year, coverage\_type, term)

premiums.append(premium)

year += 1

plt.plot(range(1, num\_years + 1), premiums)

plt.xlabel("Years")

plt.ylabel("Premium")

plt.title("Premium Over Time")

plt.show()

vehicle\_type = "car"

year = 2023

coverage\_type = "comprehensive"

term = "yearly"

num\_years = 5

generate\_premium\_chart(vehicle\_type, year, coverage\_type, term, num\_years)

