PRN: 21070521024

NAME: Dev Chadalwada

SUBJECT: GEN AI

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%.

CODE:

class VehicleInsurance:

def \_\_init\_\_(self, vehicle\_value, insurance\_years, monthly\_rate):

self.initial\_value = vehicle\_value

self.years = insurance\_years

self.monthly\_rate = monthly\_rate

def calculate\_depreciation(self, year):

depreciated\_value = self.initial\_value \* ((1 - 0.07) \*\* year)

return depreciated\_value

def premium\_chart(self):

chart = []

for year in range(1, self.years + 1):

yearly\_value = self.calculate\_depreciation(year)

yearly\_premium = yearly\_value \* self.monthly\_rate \* 12

quarterly\_premium = yearly\_premium / 4

monthly\_premium = yearly\_premium / 12

chart.append({

"Year": year,

"Depreciated Vehicle Value": round(yearly\_value, 2),

"Monthly Premium": round(monthly\_premium, 2),

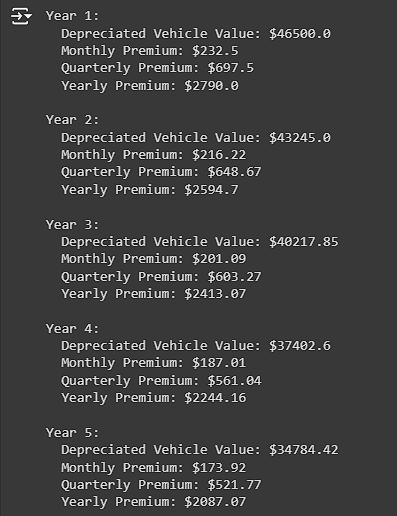
"Quarterly Premium": round(quarterly\_premium, 2),

"Yearly Premium": round(yearly\_premium, 2)

})

return chart

OUTPUT:



**Explanation:**

* **Depreciation**: The value of the vehicle depreciates by 7% every year.
* **Premium Calculation**: The premiums (monthly, quarterly, and yearly) are calculated based on the depreciated vehicle value at the beginning of each year.

Q:4 Generate a model to represent interest calculations of a Bank account where the process of calculating interest for 6 months is a. Find minimum balance for each month b. Make a total of all minimum balances c. Calculate interest based on interest rate d. Divide interest by 12 to find one-month interest e. Multiply interest by 6 to show interest in the account. Generate a model to represent transactions and interest calculations for 6 months.

CODE:

class BankAccount:

def \_\_init\_\_(self, initial\_balance, interest\_rate):

self.initial\_balance = initial\_balance

self.interest\_rate = interest\_rate

self.transactions = {i: [] for i in range(1, 7)} # Stores transactions for 6 months

def add\_transaction(self, month, amount):

if 1 <= month <= 6:

self.transactions[month].append(amount)

else:

print("Invalid month. Please enter a month between 1 and 6.")

def calculate\_minimum\_balance(self, month):

balance = self.initial\_balance

min\_balance = balance

for transaction in self.transactions[month]:

balance += transaction

if balance < min\_balance:

min\_balance = balance

return min\_balance

def calculate\_interest(self):

total\_min\_balances = 0

for month in range(1, 7):

min\_balance = self.calculate\_minimum\_balance(month)

total\_min\_balances += min\_balance

total\_interest = (total\_min\_balances \* self.interest\_rate) / 100

monthly\_interest = total\_interest / 12

six\_month\_interest = monthly\_interest \* 6

return {

"Total Minimum Balances": round(total\_min\_balances, 2),

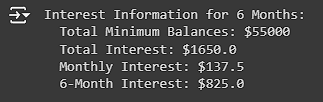
"Total Interest": round(total\_interest, 2),

"Monthly Interest": round(monthly\_interest, 2),

"6-Month Interest": round(six\_month\_interest, 2)

}

OUTPUT:



**Explanation:**

1. **Transactions**: We record deposits and withdrawals in each month.
2. **Minimum Balance Calculation**: For each month, the minimum balance is calculated after each transaction.
3. **Interest Calculation**:
   * Total interest is based on the sum of the minimum balances for 6 months.
   * The interest rate is applied to this sum to calculate the total interest.
   * We then calculate monthly interest (by dividing by 12) and 6-month interest (by multiplying by 6).