Technical Assessment

Explanation of the Code Ankith Gowda Y S

def generate\_monthly\_bill(item\_list, target\_month):

→A function(generate\_monthly\_bill) that takes two parameters (list\_items and target\_month) to calculate the bill.

year, month = map(int, target\_month.split('-'))

→ Splits the target month string (e.g., '2024-11') into year and month as integers.

month\_start = datetime(year, month, 1)

→ Creates a date for the first day of the target month.

month\_days = calendar.monthrange(year, month)[1]

→ Finds the number of days in the target month.

month\_end = datetime(year, month, month\_days)

→ Creates a date for the last day of the target month.

result = { 'line\_items': [], 'total\_revenue': 0.0 }

→ Initializes the result with an empty list of items and total revenue as 0.

for item in item\_list:

→ Loops through each item in the list.

start = datetime.strptime(item['start\_date'], '%Y-%m-%d')

→ Converts the item's start date from string to date.

end = datetime.strptime(item['stop\_date'], '%Y-%m-%d')

→ Converts the item's end date from string to date.

if end < month\_start or start > month\_end:

→ Checks if the item is not active in the target month.

continue

→ Skips this item if it's not active in the target month.

active\_start = max(start, month\_start)

→ Chooses the later date between the item's start and the month start.

active\_end = min(end, month\_end)

→ Chooses the earlier date between the item's end and the month end.

active\_days = (active\_end - active\_start).days + 1

→ Calculates how many days the item was active in the target month.

qty = int(item['qty'])

→ Gets the quantity of the item.

rate = float(item['rate'])

→ Gets the rate (price per month) of the item.

prorated\_amount = round(qty \* rate \* (active\_days / month\_days), 2)

→ Calculates the amount for the item based on active days.

* qty \* rate is the full monthly charge.
* Multiply by (active\_days / month\_days) to prorate for partial months.

result['line\_items'].append({

→ Adds the billing details of the item to the result.

'item\_code': item['item\_code'],

→ Includes the item name or code.

'rate': rate,

→ Includes the rate per item.

'qty': qty,

→ Includes the quantity.

'amount': prorated\_amount,

→ Includes the calculated charge.

'billing\_period': f"{active\_start.date()} to {active\_end.date()}"

→ Includes the date range the item was active in this month.

})

→ End of the dictionary for this item.

result['total\_revenue'] += prorated\_amount

→ Adds this item's amount to the total.

result['total\_revenue'] = round(result['total\_revenue'], 2)

→ Rounds the total revenue to 2 decimal places.

return result

→ Returns the final billing summary.

**Call function**

monthly\_bill = generate\_monthly\_bill(item\_list, "2024-11")

* Above line calls the function (generate\_monthly\_bill).
* Here we passed item\_list and month as a parameter to the function.

from pprint import pprint

* Here I imported pretty print function in order to print clean and readable format.

pprint(monthly\_bill)