**Problem Overview**

The goal of this task is to generate a monthly bill for a list of items, considering the quantity, rate, and active period for each item. The challenge is to prorate the amount based on how many days the item is active during the target month. We need to calculate the overlap between the item's active period and the month and then prorate the amount for that period.

**Approach & Solution**

**1. Parsing the Input Data**

Each item in the input data is represented as a dictionary containing the following details:

* **item\_code**: A unique identifier for the item.
* **qty**: The quantity of the item.
* **rate**: The rate at which the item is billed.
* **start\_date**: The start date of the item’s active period.
* **stop\_date**: The end date of the item’s active period.

We begin by parsing the start\_date and stop\_date from string format into Python datetime objects. This allows us to compare and manipulate the dates more easily.

**2. Calculating Active Days**

To calculate how many days an item is active in the target month, we compare the item’s active period with the target month’s date range:

1. **Finding the Overlap**: We check the overlap between the item’s active period and the month.
2. **Active Days**: If the item is active during the month, we calculate how many days it was active within the given month.

If the item is not active during the target month, we skip it.

**3. Prorating the Amount**

Once we have the number of active days for the item, we calculate the prorated amount using the following formula:

Prorated Amount=(Active Days​/ Total Days in Month)×(Rate×Qty)

Where:

* **Active Days**: The number of days the item was active in the target month.
* **Total Days in Month**: The total number of days in the target month.
* **Rate**: The rate at which the item is billed.
* **Qty**: The quantity of the item.

**4. Grouping and Summarizing the Results**

Items with the same item\_code, billing period and rate are grouped together. For each group, we sum the quantities and calculate the total prorated amount.

**5. Final Output**

After processing all the items, we produce a summary that includes:

* A list of **line items**, with details like:
  + item\_code: Unique identifier for the item.
  + qty: The total quantity of the item billed during the month.
  + rate: The rate at which the item is billed.
  + amount: The prorated amount for the billing period.
  + billing\_period: The date range for the billing period.
* The **total revenue** for the month, which is simply the sum of all prorated amounts.

**How It Works:**

**Example Test Case**

Let’s walk through a test case where the input items are:

1. A desk with a start date of November 1st, 2023, and an end date of October 17th, 2024.
2. Another desk with a start date of October 18th, 2024, and an end date of October 31st, 2025.

For each item, the program calculates the overlap with the target month (e.g., November 2024), prorates the amount based on the active days, and generates the final bill for that period.

**Conclusion**

In this solution, we efficiently handle prorating based on the active days within the month. The program calculates overlapping dates, prorates amounts, and groups items with the same rate and item code. It ensures that the results are accurate and easy to understand.

This approach is robust and can handle edge cases such as items with no active days in the month or items that span multiple months. The code is well-structured and clearly handles the main task of calculating monthly bills.