Name : Awalif Ghotamicha Irfan Saputra

- **Problem-Solving and Algorithmic Thinking**

  Scenario: The business team needs to import a CSV file containing over 100,000 product records. During testing, the import process caused high server load, timeouts, and occasional failures

  - **Identify the Problem**

    When trying to import from a CSV file, several issues can happen :

    - The app tries to read and insert all rows in a single req, which overloads the server and causes timeouts or memory crash
    - Each row inserted one by one, not using batch inserts
    - No early validation so when one row is bad or error, it could crash the whole process
    - Import runs on the main thread it could block other users from using the app smoothly
  - **Propose a Solution**

    I solve this problem by :

    - Batching or chunking CSV file, for example 2000 rows per batch
    - Using bulk insert instead of inserting one row at a time
    - When user upload CSV file, process it first in background using for example Laravel Queues
    - Make a quick validation when upload file then deeper validation inside the background job
  - **Plan the Implementation**

    To validate and keep things consistent without block the main app :

    - Offload the processing to queue job, thats make main app still responsive
    - Showing a status or progress bar to let users know, the import is in progress or not
    - Process row by row in each chunk, so when some rows fail maybe, it doesnt stop the whole import
  - **Prepare for Growth**

I solve this by :

    - Use dedicated queue workers like Redis and Laravel Horizon to process imports faster and parallel
    - Using message queues like Kafka
    - Break down import into smaller batches
    - Maybe run large imports during off hourss to reduce system load

  - **Some Tools or Packages I Use to Solve (Situation i use PHP)**
    - maatwebsite/excel for handling CSV with chunking
    - Laravel Queues for async background jobs
    - Laravel Storage for uploaded CSV
    - Logging to inform users about the result

- **Part 2: Soft Skills and Teamwork**
  - **Workload Management**

    I will focus on identifying which features are critical for delivery, such as core API functions or anything that related to the main user journey. After that, break down large tasks into smaller parts, from there we can track progress faster and make things feel more manageable
  - **Team Support**

    First, i try to talk to them and ask where they are stuck. If i can help directly like maybe reviewing their code, helping to debug i will jump in. But if they are overloaded, i will suggest redistributing the task among the team, especially if some members have a bit more room. The most importan is keep the mood positive and never blame each other
  - **Communication with Stakeholders**

    I will be transparent then i try to explain about what caused the delays, what we have done so far something like that. If possible, i will offer a partial delivery so stakeholders still see the progress. My princip is, clear communication make trust.