Task: Build a Reporting Module Using MySQL Only

Objective

Create a reporting feature that gathers data from different MySQL tables, applies filters, combines the required information, and shows it in a clean, paginated report. The module must be fast, efficient, and avoid loading unnecessary data.

Task Details (Theoretical Only)

1. Read Data From Multiple Tables

- The module must pull information from different MySQL tables such as:
  - Transactions
  - Users
  - Games
  - Providers
- Each section of data must work together to form a complete report.

---

2. Efficient Data Gathering

The report should NOT fetch all rows from tables.

Expected working:

- Only collect what is needed based on user-selected filters.
- Avoid unnecessary data processing.
- Keep memory usage low.

3. Apply Common Report Filters

The user should be able to filter the report using:

- Date range (from–to)
- Status (success, failed, pending, etc.)
- User ID or Username
- Game/Provider selection
- Amount range (min–max)

These filters must refine the report before showing it.

4. Combine the Data Into One Report

Even though the information comes from multiple tables, the final output should be shown as one unified report.
The merging or combining logic must ensure:

- Clean structure
- No duplicated entries
- No missing data

5. Add Pagination

The final output must not show everything at once.

The module must:

- Show only a limited number of rows per page
- Allow user to navigate page by page
- Keep the results smooth and fast even when the dataset is large

---

6. Cache the Final Report

Once the report for a set of filters is generated:

- Save/store the result temporarily
- So when the user requests the same filters again, the system returns instantly
- Refresh cache periodically

This helps reduce repeated work and improves user experience.

---

7. Final Output Should Contain

The report must include:

- Total number of records
- Current page
- Total pages
- Records of the current page
- Which filters were applied

---

8. Stability and Error-Free Handling

The module must:

- Handle empty results cleanly
- Handle invalid filters
- Handle missing data
- Never break or crash the system

---

9. Documentation Required

A senior developer must provide:

- Overview of how the report works
- How filters are handled
- How the data is combined
- How pagination is managed
- How caching works
- Any limitations or future improvements

In Simple Words

You are creating a **smart reporting module** where:

- Data comes from several MySQL tables
- You filter only what you need
- Combine everything into one report
- Show the report page-by-page
- Save the final result for faster next-time loading