**Question 1:** What is the difference between Brainstorming and JAD Sessions?

Both **Brainstorming** and **JAD (Joint Application Development) Sessions** are techniques used in gathering requirements and generating ideas, but they’re quite different in structure and purpose.

|  |  |
| --- | --- |
| **Brainstorming** | **JAD Sessions** |
| Purpose: Generate as many ideas as possible, often creatively and freely. | Purpose: Gather detailed, structured requirements for a system through collaboration. |
| Style: Informal, open and unstructured discussion to encourage creativity. | Style: Highly structured, formal sessions with a clear agenda and defined roles. |
| Participants: Usually a small group of stakeholders, team members or experts. | Participants: Business users, subject matter experts, IT staff and a facilitator. |
| Duration: Typically short (a few hours or a day). | Duration: Can span multiple days or weeks depending on project scope. |
| Output: List of ideas, possible solutions, concepts or high-level requirements. | Output: Documented and agreed-upon detailed system requirements. |
| Focus: Creativity and idea generation (no bad ideas at this stage). | Focus: Clarifying and validating requirements; resolving conflicts. |

**Question 2:** Why Document Analysis is one of the compulsory technique we use in a Project? Justify

**Document Analysis** is considered a compulsory or at least highly recommended technique in most projects because it gives you a solid foundation to understand the current state, the business rules, and the expectations even before interacting with people.

**Why Document Analysis is Essential in Projects**

1. Existing Knowledge is Already Captured

* In most organizations, valuable information is already documented — whether it’s in the form of business process documents, SOPs, policy manuals, contracts, user manuals, technical specifications, audit reports or previous project documentation.
* Ignoring these would mean reinventing the wheel or missing critical background.

2. Helps Understand the Current State (AS-IS)

* Before suggesting new requirements or designing a new system, you must understand how the business currently operates.
* Document analysis helps you map the “AS-IS” process without wasting time asking for basic information during stakeholder interviews.

3. Identifies Hidden or Unstated Requirements

* Often, stakeholders forget to mention certain rules or processes simply because they assume "everyone knows" — but these are usually captured in documents.
* Example: A policy document might reveal security or compliance rules that the users didn’t mention during discussions.

4. Saves Time in Stakeholder Meetings

* By studying documents first, you approach interviews and workshops more prepared.
* You can ask more precise, meaningful and advanced questions rather than starting from scratch.

5. Ensures Accuracy & Completeness

* Relying only on verbal communication (like interviews) can lead to misunderstandings or forgotten details.
* Documents offer a verifiable and stable reference.

6. Legal and Compliance Validation

* For many projects (especially in finance, healthcare, government, and CSR-related systems), legal and compliance requirements are written in contracts and policies.
* Skipping document analysis can result in violations, which could be costly.

**Question 3:** In Which Context we will use Reverse Engineering?

Reverse engineering is the process of analysing an existing system, application or product to understand its design, architecture, code and functionalities when **documentation is missing, outdated or incomplete.**

**Contexts Where Reverse Engineering is used**

|  |  |
| --- | --- |
| **Context** | **Explanation** |
| Legacy System Modernization | When the company has an old system but little or no documentation. Reverse engineering helps understand how the system works before replacing or upgrading it. |
| Integration Projects | When you need to connect a new system to an old one, and the existing system’s design isn’t documented — reverse engineering helps you uncover the APIs, data structures, and workflows. |
| Software Migration | When moving from one platform to another, reverse engineering helps understand dependencies and architecture of the old system. |
| Competitor Analysis / Product Deconstruction | In some cases (ethically, and within legal limits), reverse engineering is done to understand how a competitor’s product works, especially for benchmarking or compatibility. |
| Security Assessment | Security experts often reverse-engineer software to detect vulnerabilities or malicious code when source code is unavailable. |
| Bug Fixes in Unmaintained Software | When no active support or documentation is available for an old piece of software, reverse engineering helps developers identify and fix issues. |

**Question 4:** What is the difference between Brainstorming and Focus Groups?

|  |  |
| --- | --- |
| **Brainstorming** | **Focus Groups** |
| Purpose: Generate a wide range of ideas, solutions, or options — encourage creative, free thinking. | Purpose: Collect opinions, attitudes, and feedback on a product, service or concept from a group of users. |
| Style: Unstructured or semi-structured, open idea-sharing; quantity of ideas is encouraged over evaluation. | Structured and moderated discussion; the moderator guides the group through specific questions or topics. |
| Participants: Typically internal team members, stakeholders, or subject matter experts. | Typically actual or potential users, customers or target audience representatives. |
| Focus: Idea generation (solutions, new features, alternatives). | Opinion gathering (likes, dislikes, experiences, perceptions). |
| Goal: Create innovative options and explore possibilities. | Understand user expectations, needs, and attitudes. |
| Output: A list of creative ideas, concepts, or possible requirements. | Validated insights, preferences, concerns, and real-world feedback. |
| Timing: Early in the requirement elicitation or problem-solving stage. | Usually after some initial ideas are formed, or when evaluating a prototype or concept. |

**Question 5:** Observation Technique – Explain both Active and Passive approaches

Observation is a requirement elicitation technique where the Business Analyst or observer **watches users perform their tasks** in the real environment to understand their workflows, challenges, and needs.

**Two Approaches: Active vs. Passive Observation**

|  |  |
| --- | --- |
| **Active Observation** | **Passive Observation** |
| Definition: The observer participates in the process and sometimes interacts with the user while observing. | Definition: The observer quietly watches the user perform tasks without interrupting or influencing them. |
| Interaction: High — the observer may ask questions, seek clarification or even try the task themselves. | Interaction: None — purely observing real-time actions as they happen. |
| Purpose: To gain deeper insights, clarify doubts immediately, and understand the reasoning behind user actions. | Purpose: To capture unbiased, natural behavior and process flow without any interference. |
| When to Use: When the system or task is complex, or when real-time clarification is needed. | When to Use: When you want to understand true user behavior or test whether processes are followed as documented. |
| Example: A Business Analyst works alongside a warehouse employee, asking questions about why they scan products a certain way. | Example: A Business Analyst watches a cashier process transactions for an hour without saying anything, just noting their actions. |

**Question 6:** How do you conduct the Requirements Workshop?

**Requirements Workshops** are one of the most effective and collaborative ways to gather and refine requirements in any project

**Steps to Conduct a Requirements Workshop**

**1. Plan the Workshop**

Define Objectives: What is the workshop aiming to achieve? e.g., identify new features, finalize business rules and prioritize requirements.

Select Participants: Invite the right stakeholders:

* Business users
* Subject matter experts (SMEs)
* Technical team representatives (developers, testers, architects)
* Product Owner or Project Sponsor
* Facilitator (usually the Business Analyst!)

**2. Prepare the Agenda**

* Create a clear agenda and share it in advance.
* Break it into focused sections like:
  + Introduction & goals
  + Current state review
  + Requirement discussions
  + Open Q&A
  + Summary & next steps.

**3. Set the Environment**

Arrange a suitable space with:

* Whiteboards or collaboration tools (e.g., Miro, Lucid chart, MS Teams, Zoom breakout rooms).
* Required documents (previous specs, business rules, user stories and wireframes).

**4. Facilitate the Workshop**

* Start by reviewing the objectives and agenda.
* Encourage open discussion, but stick to the topic.
* Use elicitation techniques during the session:
  + Brainstorming
  + Document review
  + Use case walkthroughs
  + Prototyping or visual aids
* Resolve conflicts (if any) on requirements or priorities.
* Keep detailed notes, decisions, and action items.

**5. Document the Outcomes**

* Prepare a workshop summary document or meeting minutes and circulate it to all participants for review and confirmation.

**6. Follow-Up**

* Update the requirements documents (BRD, SRS, User Stories).
* Arrange follow-up sessions if there were unresolved issues.
* Seek sign-off from key stakeholders.

**Question 7:** In which context, Interview Technique can be conducted by a BA? How may approaches are there in conducting Interviews? (Structured – Unstructured) Explain them. Explain the difference between Open Ended Questions and Closed ended Questions

**When is the Interview Technique used by a Business Analyst?**

A Business Analyst conducts interviews when:

* Direct communication with stakeholders, users or Subject Matter Experts (SMEs) is needed.
* Requirements need clarification, validation or detailed exploration.
* There are complex or sensitive topics that require one-on-one attention.
* Stakeholders are not comfortable in group settings (like workshops or focus groups).
* You need to gather insights, personal experiences or opinions.

**Approaches to Conducting Interviews**

|  |  |
| --- | --- |
| **Approach** | **Description** |
| Structured Interview | The interviewer prepares a fixed set of questions in advance. All participants are asked the same questions in the same order. Useful when you need standardized data. |
| Unstructured Interview | The interviewer has no formal question list — the conversation flows naturally based on the participant's responses. Useful for deep exploration of ideas. |
| Semi-Structured Interview (Common in projects) | A mix of both — the BA prepares key questions but allows flexibility to ask follow-ups or go off-script when interesting insights emerge. |

**Difference between Open-Ended and Closed-Ended Questions**

|  |  |
| --- | --- |
| **Open-Ended Questions** | **Closed-Ended Questions** |
| Encourage the person to describe, explain or elaborate. | Expect a short, specific, factual answer (usually "Yes/No" or fixed options). |
| Used to gather opinions, feelings, reasons and insights. | Used to gather specific data or confirmations. |
| Start with: "What," "Why," "How," "Describe," or "Explain". | Start with: "Is," "Are," "Do," "Did," "Have," "Can." |
| Example: “How do you handle order cancellations?” | Example: “Do you allow order cancellations? Yes or No?” |

**Question 8:** Questionnaire Technique – Where we will use? Give one example

**The Questionnaire Technique is used when:**

* You need to collect information from a large number of people in a short time.
* Stakeholders are spread across different locations or have limited availability for face-to-face meetings.
* You want standardized responses that are easy to analyze (especially when opinions, preferences, or usage patterns are needed).
* You are doing surveys, user feedback, or requirement validation.

**Where to Use Questionnaire Technique:**

* During requirement elicitation to gather needs from many users.
* During requirement validation to confirm stakeholder agreement.
* During market research or feasibility studies to understand customer expectations.
* During user acceptance planning to assess readiness and feedback.

**Example Scenario:**

**Project:** Online Agriculture Product Store.

**Context:** You want to gather feedback from 100 farmers and customers about their preferred payment methods and delivery options.

**Sample Questions:**

**Closed-ended:** Which payment method do you prefer?

[ ] Credit/Debit Card

[ ] UPI

[ ] Cash on Delivery

[ ] Net Banking

**Open-ended:** What challenges do you face while ordering farm supplies online?

**Question 9:** How to Sort the Requirements – Where we will use? Give one example

**Sorting requirements** is a very important step in Business Analysis and is commonly used during the **Requirement Analysis** phase to organize and manage requirements effectively.

**How to Sort Requirements**

**Sorting means organizing the gathered requirements based on specific criteria such as:**

|  |  |
| --- | --- |
| **Criteria** | **What it Means** |
| **Business Priority** | **Which requirements are most critical for business goals?** |
| **Type of Requirement** | **Functional, Non-Functional, Technical, UI, etc.** |
| **Stakeholder** | **Who requested it — HR, Accounts, Employee, etc.?** |
| **Phase of Implementation** | **To be delivered in Phase 1, 2 or future releases.** |
| **Feasibility / Complexity** | **How easy or hard it is to implement (High/Medium/Low)?** |
| **Legal / Compliance Impact** | **Must-have vs Nice-to-have due to policy/legal reasons.** |

**We use it when:**

* **Creating the Requirement Traceability Matrix (RTM)**
* **Preparing Business Requirement Document (BRD)**
* **Doing Prioritization & Planning (e.g., in Agile sprints)**
* **During Release Planning / Phased Rollouts**

**Example:**

Suppose you've gathered few requirements for the Employee Loan Management System.

You can **sort by priority** like below:

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Priority** | **Phase** |
| Employees should be able to apply for a loan. | High | Phase 1 |
| HR should review and approve/reject loans. | High | Phase 1 |
| System should auto-deduct loan EMIs from salary. | High | Phase 1 |
| Employees can view their loan status. | Medium | Phase 1 |
| Dashboard with charts for Management | Low | Phase 2 |
| Mobile App version | Low | Phase 3 |

**Question 10:** Prioritise the Requirements – Where we will use? Give one example

**Prioritization of requirements** is the process of ranking requirements based on their importance, urgency, and business value, so that the most critical ones are developed and delivered first.

**Where Do We Use Prioritization?**

* During Requirement Analysis — after gathering all requirements, to decide the order of development.
* During Sprint or Iteration Planning — to select high-priority features for the upcoming development cycle.
* During Scope Management — to handle situations when time, budget, or resources are limited.
* During Change Requests Evaluation — to see if new requests should replace or push back existing items.

**Common Prioritization Techniques:**

* MoSCoW Method: Must Have, Should Have, Could Have And Won’t Have.
* Kano Model: Customer satisfaction based categories.
* 100-Dollar Test / Voting: Stakeholders distribute points or virtual money to rank the features.
* Business Value vs. Complexity Matrix: High-value, low-effort items are prioritized first.

**Example Scenario:**

**Project**: Online Agriculture Product Store.

After collecting the following requirements, Priority is given based on MoSCoW method.

|  |  |
| --- | --- |
| **Requirement** | **Priority** |
| User Login & Registration | Must Have |
| Secure Payment Integration | Must Have |
| Product Search & Filter | Must Have |
| Product Rating & Reviews | Should Have |
| Social Media Sharing | Could Have |
| Wish List Functionality | Could Have |

**Question 11:** Weekly status reporting – How we will drive?

**Weekly Status Reporting** is a key part of project tracking and communication.

**How to Drive Weekly Status Reporting**

**1. Set a Fixed Schedule**

* Decide the day and time (e.g., Every Friday by 5 PM or Monday morning).
* Consistency builds trust with stakeholders.

**2. Define the Report Format**

A standard weekly status report usually includes these sections:

|  |  |
| --- | --- |
| **Section** | **Purpose** |
| Project Name & Date | Identifies the report. |
| Reporting Period | Covers which week (e.g., April 14–April 20). |
| Overall Status | Green / Yellow / Red (visual status). |
| Accomplishments This Week | What was completed (tasks, milestones). |
| Planned for Next Week | Upcoming tasks or goals. |
| Issues / Risks | Challenges, blockers, or delays. |
| Dependencies | Anything waiting on external inputs. |
| Action Items & Owners | Who is responsible for pending tasks? |

**3. Gather Input from the Team**

* Get updates from Developers, Testers, Project Manager and Stakeholders.
* This ensures the report reflects the real project status, not just assumptions

**4. Summarize and Share**

* Keep it clear, concise and visual using tables, charts and color codes.
* Share with Project Sponsor, Stakeholders, Development & QA teams and your manager or clients.

**5. Discuss in Status Meetings**

* Use the report as the main discussion document during the weekly status call.
* Address - What’s progressing well, what’s blocked or delayed and what decisions are needed.

**Question 12:** Meeting Minutes Document – prepare one Sample

A **Meeting Minutes Document** is an official written record of what was discussed, decided, and assigned during a meeting.

In meeting we captures Key points discussed, Decisions made, Action items and Next meeting schedule

**Purpose of Meeting Minutes:**

* Ensure everyone has a clear reference for what was agreed.
* Help those who missed the meeting to catch up.
* Track action items and responsibilities.
* Provide an audit trail of discussions and decisions.
* Avoid misunderstandings or future disputes.

**Typical Format of Meeting Minutes:**

|  |  |
| --- | --- |
| **Section** | **Description** |
| Project Name | Name of the project. |
| Meeting Title | Purpose or topic of the meeting. |
| Date & Time | When the meeting took place. |
| Facilitator / Note Taker | Who led the meeting and who recorded notes. |
| Attendees | List of participants. |
| Agenda | Topics planned for discussion. |
| Discussion Points | Summary of what was actually discussed. |
| Decisions Made | Agreed solutions or outcomes. |
| Action Items | Task list with assigned owners and deadlines. |
| Next Meeting | When the next meeting is scheduled. |

**Example / Sample:**

**Project Name:** Online Agriculture Product Store

**Meeting Title:** Weekly Status Update

**Date:** April 22, 2025

**Time: 3:00 PM – 4:00 PM**

**Location: MS Teams Virtual Meeting**

**Facilitator:** Business Analyst Name

**Note Taker:** Venkatesh Dhanana (Can be BA)

**Attendees:** Project Sponsor, Project Manager, Sr. Java Developer, Stakeholder, BA

**Agenda:**

* Review current development status.
* Discuss payment gateway integration issues.
* Finalize product categorization for the catalog.
* Next steps and action items.

**Discussion Points:**

* Development progress on the payment module.
* Risks due to late API integration.

**Decisions Made:**

* Payment vendor escalation required.

**Action Items:**

|  |  |  |
| --- | --- | --- |
| Task | Owner | Deadline |
| Follow up with vendor | Project Manager | April 22 |

**Next Meeting:**

* Date: April 27, 2025
* Time: 3:00 PM – 4:00 PM
* Location: MS Teams

**Question 13:** Change Tracker – Document – prepare one Sample

This type of document is very useful to track changes in requirements, scope, design, or any agreed deliverables, especially in ongoing projects like your **Online Agriculture Product Store**.

* The Change Tracker ensures that every change is:
  + Documented.
  + Evaluated.
  + Approved.
  + Assigned for implementation.
  + Tracked to closure.

**Change Tracker Document — Sample**

* **Project Name:** Online Agriculture Product Store
* **Document Name:** Change Tracker
* **Prepared By:** [BA Name]
* **Date:** April 22, 2025
* **Version:** 1.0

**Change Log Table**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Change ID** | **Change Description** | **Raised By** | **Date Raised** | **Impact Analysis** | **Approval Status** | **Owner** | **Target Closure Date** | **Actual Closure Date** |
| CHG-001 | Add "Organic Fertilizers" as a new product category. | Mr. Peter (Farmer) | April 18, 2025 | Medium: Update to DB & UI. | Approved | BA) | April 21, 2025 | Pending |
| CHG-002 | Adjust delivery window from 2 days to 3 days for remote areas. | Mr. Henry (Sponsor) | April 15, 2025 | High: Impacts logistics module. | Pending Approval | Mr. Vandanam (PM) | April 25, 2025 | N/A |
| CHG-003 | Modify payment flow for split payment option. | Ms. Juhi (Dev) | April 17, 2025 | High: Requires design & testing update. | Approved | Dev Team | April 30, 2025 | In Progress |

Note:

* All changes must be reviewed and impact-analysed before implementation.
* Approved changes are updated in the Requirements Document and Project Plan.
* This document is shared weekly with stakeholders.

**Question 14:** Difference between Traditional Development Model and Agile Development Models.

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Traditional Development Model (Waterfall)** | **Agile Development Model** |
| Approach | Sequential / Linear — each phase must finish before the next starts. | Iterative & Incremental — work is done in short cycles (Sprints). |
| Flexibility to Change | Low — once the requirements are frozen, changes are difficult and costly. | High — changes are welcomed, even late in the process. |
| Requirements Gathering | Done upfront — detailed and complete before development starts. | Continuous — requirements evolve during the project. |
| Project Visibility | Limited — working product is only seen at the end of the project. | High — working software is delivered at the end of every Sprint. |
| Customer Involvement | Minimal — mostly involved at the beginning (requirements) and at the end (delivery). | High — customer is involved throughout (planning, demos, feedback). |
| Testing | Usually done after development (late testing). | Testing is done parallel to development (continuous testing). |
| Risk Handling | High risk — issues are identified late, sometimes after major rework. | Low risk — early and frequent feedback helps identify problems early. |
| Documentation Focus | Heavy documentation before coding starts. | Lightweight documentation — focus on working software. |
| Best suited for | Projects with clear, stable, fixed requirements (e.g., construction, hardware). | Projects with changing, evolving, and complex requirements (e.g., software, startups). |

**Question 15:** Explain Brainstorming Technique – Where to use?

**Brainstorming** is a group creativity technique used to **generate a large number of ideas or solutions** for a problem, business need or requirement in a short amount of time.

**1. You use Brainstorming when:**

* You need to gather new ideas or solutions from a group.
* Requirements are unclear or there are multiple possibilities.
* You want different perspectives (from developers, testers, stakeholders, end-users).
* At the early stage of projects — for requirement gathering, feature ideas, problem-solving, or risk identification.

**2. Example Scenario:**

**Project:** Online Agriculture Product Store.

**Situation:** You want to identify possible features that farmers would need on the platform.

**Brainstorming Session Output:**

* Bulk order discount option.
* Real-time delivery tracking.
* Fertilizer calculator tool based on land size.
* Voice search for illiterate users.

All these ideas are collected — and later refined, validated and prioritized.

**3. Steps in Brainstorming:**

* Define the Problem Clearly.
* Gather the Right Group (stakeholders, SMEs, developers, users).
* Set Rules (free thinking, no judgment, quantity over quality first).
* Capture All Ideas (whiteboard, sticky notes, virtual boards).
* Review and Shortlist the best ideas after the session.

**Question 16:** What reports Accounts Departments will generate

For a system like **Employees Loan Management System**, the **Accounts Department** will need clear, detailed and audit-friendly reports for monitoring and managing all financial transactions related to employee loans.

Here’s a practical list of reports the Accounts team would typically generate.

|  |  |
| --- | --- |
| **Report Name** | **Description** |
| Loan Sanction Report | List of all loans approved (Employee Name, Loan ID, Amount and Sanction Date). |
| Loan Repayment Status Report | Tracks EMI payments, due dates, paid/unpaid status and balance amount. |
| Loan Recovery Report | Summary of loans recovered via salary deductions — with amounts and dates. |
| Loan Rejection Report | Shows all rejected applications with reasons for audit and analysis. |
| Loan Outstanding Summary | Shows total loan amount disbursed vs outstanding amount per employee. |
| EMI Due Report | List of upcoming EMIs within a date range for planning deductions. |
| Loan Closure Report | List of loans fully paid and closed, along with closure dates. |
| Audit Trail Report | Full record of changes in loan applications, approvals and repayments for compliance. |

**Question 17:** What is the structure of the message/mail communicated from the HR department to the employee in case the Loan is rejected?

**Subject:** Loan Application Status — [Employee Name] — [Loan ID]

Dear [Employee Name],  
Greetings from the HR Department!

We would like to inform you that your loan application (Loan ID: [Loan ID]) submitted on [Application Date] has been carefully reviewed by the concerned departments.

After a thorough evaluation, we regret to inform you that your loan request has been declined due to the following reason(s):

Reason for Rejection:

Ex: Eligibility criteria not met or insufficient salary for the requested loan amount or incomplete documentation.

We encourage you to review the above points and, if applicable, you may resubmit your application once the identified issues are resolved.

For any clarifications or assistance, please feel free to reach out to us.

Thank you for your understanding.  
We remain committed to supporting your needs.

Best regards,  
HR Department  
TTS Company

**Question 18:** What is the structure of the message/mail communicated from the HR department to the employee in case the Loan is approved?

**Subject:** Loan Application Approved — [Employee Name] — [Loan ID]

Dear [Employee Name],  
Greetings from the HR Department!

We are pleased to inform you that your loan application (Loan ID: [Loan ID]), submitted on [Application Date], has been approved after review by the HR and Accounts departments.

Below are the key details of your loan offer:

Loan Details:

* Loan Amount: ₹ [Amount]
* Interest Rate: [Interest %] p.a.
* Repayment Period: [Number of Months]
* Monthly EMI: ₹ [EMI Amount]
* Repayment Start Date: [Date]
* Salary Deduction: Automatic deduction from monthly salary.

Attached to this email, you will find:

* Loan Approval Terms & Conditions Document
* Loan Repayment Schedule

Please review the documents carefully. If you agree to the terms and conditions, kindly send your acceptance confirmation by replying to this email, or by digitally signing the attached consent form.

Upon your confirmation, the loan amount will be disbursed and the repayment process will be initiated as per the provided schedule.

For any questions or clarification, feel free to contact the HR or Accounts team.

Congratulations and thank you!  
We are happy to support you.

Best regards,  
HR Department  
TTS Company

**Question 19:** Design a sample report on the Loans applications Received by the accounts department.

**Sample Report Format** for the **Loan Applications Received by the Accounts Department**

**TTS Company — Loan Applications Received Report**

**Department: Accounts**

**Reporting Period: [Start Date] to [End Date]**

**Prepared By: [Preparer's Name]**

**Date: [Report Date]**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S.No.** | **Loan ID** | **Employee Name** | **Employee ID** | **Department** | **Requested Amount (₹)** | **Application Date** | **Status** | **Reviewed By** | **Remarks / Reason** |
| 1 | LN2025-001 | John Doe | EMP-1001 | IT | ₹1,50,000 | 01-Apr-2025 | Approved | Ramesh A. | Approved - Eligible |
| 2 | LN2025-002 | Priya Sharma | EMP-1023 | HR | ₹2,00,000 | 03-Apr-2025 | Rejected | Sunitha P. | Salary below threshold |
| 3 | LN2025-003 | Rahul Menon | EMP-1107 | Finance | ₹75,000 | 04-Apr-2025 | Approved | Ramesh A. | Approved - Clean record |
| 4 | LN2025-004 | Swetha Rao | EMP-1033 | Marketing | ₹3,00,000 | 05-Apr-2025 | Pending | -- | Awaiting HR Review |
| 5 | LN2025-005 | Kevin Patel | EMP-1077 | IT | ₹50,000 | 06-Apr-2025 | Approved | Sunitha P. | Approved |

**Question 20:** Which reporting Tools we will use for generating reports

Popular Reporting Tools for Generating Reports

|  |  |  |
| --- | --- | --- |
| **Tool Name** | **Use Case / Features** | **Usage Level** |
| Microsoft Excel | Quick ad-hoc reports, pivot tables, loan summaries, EMI calculations. | Common |
| Power BI | Dashboards, visual insights, interactive loan reports. | Advanced |
| Tableau | Dynamic visual reports, graphs on loan trends, repayment performance. | Advanced |
| Crystal Reports (SAP) | Formal structured financial and business reports, with export options (PDF, Excel). | Enterprise |
| JasperReports | Java-based open-source reporting — integrates well with in-house applications. | Developer / Embedded |
| SQL Server Reporting Services (SSRS) | Database-driven reports directly from loan management system. | Technical / Enterprise |