Indian Institute of Technology Madras Zanzibar

School of Engineering and Science

Z2004: Semester Project Guidelines

RAG Pipeline and AI Recommendation Engine — Full Rubric

Z2004 — Database Management Systems

Document 3 of 6 | B.S. Data Science and Artificial Intelligence, Year 2

Even Semester: March – July 2026

Dr. Innocent Nyalala

Assistant Professor of Data Science and AI

Head, SAAIL Lab | IIT Madras Zanzibar

innocent@iitmz.ac.in | Â Office Hours: Wednesday 2:00–4:00 PM | + Room 106

Teaching Assistants

Support hours: **Friday 2:00–4:00 PM (EAT)**

Sri Advaita Varakavi | # zda23b010@iitmz.ac.in

Saleh Saleh | # zda23b001@iitmz.ac.in

1 Project Overview 2

2 Submission Standards (Applies to All Milestones) 2 2.1 How Marks and Scaling Work .

. . . . . 2 2.2 Deadlines and Time Zone . . . .

system. The final grade reflects what you *built*, not what you *described*.

Team Policy

You may work **solo** or in a **team of 2–3** students. No teams of 4 or more. Register your team (or solo project) and topic by **27 March 2026 (Milestone 0)**. Registrations after this date lose 10 marks from the final project score.

M0 — Registration (0%)	
M1 — Schema & DDL (20%)	15 May 2026 data.csv or data.json + queries.sql + README.md
M2 — Dataset & queries (20%)	5 June 2026 Max 5-page PDF +
M3 — Performance evidence (20%) Final submission (40%)	performance.sql 22 June 2026 GitHub link +
27 March 2026 Group form: members + track + 1-paragraph idea	5-minute demo video + Q&A
10 April 2026 ER diagram + schema.sql (runs cleanly)	

2. Submission Standards (Applies to All Milestones)

2.1. How Marks and Scaling Work

Milestone weights add up to 100% of the project mark. The project mark is then scaled into the **course grade component (20%)**.

2.2. Deadlines and Time Zone

All deadlines in this document are in **EAT** and are due by **23:59 EAT** unless stated otherwise.

2.3. Submission Method (All Milestones)

Submit via **Google Classroom**. You may upload a ZIP file and/or paste your GitHub link(s) in the Classroom submission.

2.4. Questions and Support

For questions, contact the Teaching Assistants (listed on the title page). Their support hours are **Friday, 2:00–4:00 PM (EAT)**.

2.5. Reproducibility (Required)

Your project must be reproducible by the instructor/TAs from a clean machine. At minimum, your repository/ZIP must include:

• Exact environment specification: requirements.txt (or environment.yml) • A clear README.md with step-by-step setup and run instructions

• Database setup: DDL script(s) and data import instructions (or a seed script) • One command that runs your app end-to-end (or explicit run steps) • If you rely on API keys: use .env variables and document them (never commit keys)

2.6. Data Policy: What Counts as "Meaningful" Data

Allowed: public/open datasets; your own collected data with permission; data generated by a *documented* simulator that mimics realistic constraints (e.g., consistent IDs, timestamps, distributions, referential integrity).

Not allowed: pure random rows with no real semantics; copy-paste duplicates to inflate row counts; lorem-ipsum text that does not relate to your domain.

You must document: data source, collection/cleaning steps, and any synthetic genera tion method.

2.7. AI Tools Policy (Disclosure Required)

You may use AI tools e.g., ChatGPT/Copilot for brainstorming, debugging, and code suggestions, but:

- You are responsible for correctness, security, and understanding of everything you submit.
- Do not paste large generated code without reading and adapting it.
- In your README.md (or report), include a short **AI Usage Disclosure** listing: tool name, what it was used for (e.g., "helped draft SQL query"), and what you changed.

3. Late Submission, Extensions, and Demo Contingencies

Unless an extension is approved in writing **before** the deadline, late work is penalized at **40% of that milestone's marks per calendar day**, up to **2 days**. After 2 days, submissions receive zero for that milestone.

Extensions: Request at least 24 hours in advance (except emergencies). Provide a short reason and a revised plan. Approval is at instructor discretion.

Live demo contingencies: If your demo fails due to setup issues (missing

dependencies, DB not running, no sample user, etc.), you may be asked to continue with instructor guidance, but marks may be reduced. If there is a documented emergency, a reschedule may be granted.

4. Approved Project Tracks

4.1. Track A: RAG Pipeline (Retrieval-Augmented Generation)

Build a question-answering system backed by a relational database and a vector store. **Minimum Technical Requirements:**

- A normalised relational database (PostgreSQL *or* SQL Server; minimum 4 tables, minimum 3NF)
- At least 1000 rows of meaningful data (see data policy above)
- Text embeddings generated using a sentence-transformers model •

Vector storage in pgvector (PostgreSQL) or Pinecone (cloud)

- A Python interface that accepts a natural-language question and returns a grounded answer with source citations
- At least one B-Tree index with execution-plan evidence of use (e.g., EXPLAIN ANALYZE)

Note: If you use SQL Server for the relational database, use Pinecone (or another external vector store). pgvector is PostgreSQL-only.

4.2. Track B: AI Recommendation Engine

Build a product or content recommendation system powered by a relational backend. **Minimum Technical Requirements:**

- A normalised PostgreSQL or SQL Server database (minimum 5 tables, 3NF)
- At least 2000 rows of transactional data
- Collaborative filtering implemented using SQL window functions (ROW_NUMBER, RANK, SUM OVER PARTITION BY)
- A Python API that returns top-N recommendations for a given user •

A performance benchmark: query time with and without indexes

- A stored procedure or trigger that updates recommendation scores on new data

4.3. Track C: Advanced Schema and Analytics Platform

Design a complex multi-domain database and build an analytics dashboard. **Minimum Technical Requirements:**

- A normalised schema with minimum 7 tables spanning at least 2 domains
- All four normal forms (1NF–BCNF) demonstrated and documented

- A suite of 20 SQL queries (DDL, DML, aggregation, joins, subqueries, CTEs, window functions, transactions)
- One recursive CTE and one stored procedure
- A Python pandas dashboard (Jupyter Notebook) with at least 5 visualisations •

Query performance comparison: before and after indexing strategy

5. Milestone Descriptions and Rubrics

5.1. Milestone 0 — Group Registration (0%, Mandatory)

Due: 27 March 2026, 23:59 EAT
Submission: Google Classroom
Contents: Group member names and student roll numbers; chosen project track; one paragraph project description (domain, data source, problem statement). **Checklist (must be obvious in the submission):**

- Solo or team of 2–3 students (no teams of 4+)
- One selected track (A/B/C)
- Clear data source (where it comes from) and what you will store in the database

Groups that do not submit Milestone 0 by the deadline are not eligible to submit any subsequent milestones and receive zero for the entire project.

5.2. Milestone 1 — Schema Design and DDL (20% of project)

Due: 10 April 2026, 23:59 EAT
Submission: ZIP containing: ER diagram (PNG/PDF) + schema.sql

Checklist (minimum):

- ER diagram readable at 100% zoom (labels not blurry)
- Keys and cardinalities shown; all relationships justified
- 3NF argument: functional dependencies and decomposition notes
- schema.sql runs from scratch with no manual edits
- Constraints included: PK, FK, UNIQUE, CHECK (where appropriate)

5 ER Diagram Quality Complete, correct, all entities/relationships/attributes shown
5 Normalisation Tables are demonstrably in 3NF; functional dependencies listed
5 DDL Script Runs without errors in SQL Server or PostgreSQL; all constraints present
5 Annotations SQL comments explain design decisions in each table **20 Total (scaled to 20%)**

5.3. Milestone 2 — Dataset and Queries (20% of project)

Due: 15 May 2026, 23:59 EAT
Submission: ZIP: data.csv or data.json + queries.sql + README.md

Checklist (minimum):

• Minimum row counts met (Track A: 1000+; Track B: 2000+; Track C: sufficient for analysis)

• Data dictionary: column meanings, units, allowed ranges (in README) • Import steps: exact command(s) used to load data into PostgreSQL/SQL Server • queries.sql contains the required variety and each query is labelled • All queries run on a fresh database with your provided scripts/data

10 Dataset Quality Real or realistic; minimum row counts met; no junk data **10 Query Coverage** At minimum: 2 aggregations, 2 joins, 2 subqueries, 2 CTEs, 2 window functions
10 Query Correctness All 10 queries return correct, meaningful results **5 Code Quality** Queries are formatted, commented, and named descriptively
5 README Clear instructions to reproduce results from scratch **40 Total (scaled to 20%)**

5.4. Milestone 3 — Performance Evidence (20% of project)

Due: 5 June 2026, 23:59 EAT
Submission: PDF report (max 5 pages) + performance.sql

What the PDF must contain (minimum): a short summary of your project; objective; methodology; results; interpretation of results; relevant visualisations/tables; conclusions.
Checklist (minimum):

• Clearly state your test environment: DB engine/version, machine specs, dataset size • Identify 2–3 slow queries and justify why they matter

• Provide before/after timing method (same dataset, same conditions) • Include query plans (e.g., EXPLAIN ANALYZE output) before and after indexes • Provide the exact index DDL used and why you chose each index

• One working stored procedure or trigger (what it does and how to run it)

10 Index Design Correct index types chosen with written justification
10 EXPLAIN ANALYZE time recorded
Before/after evidence; query plans attached;

10 Interpretation Student correctly reads and explains the execution plan **10 Stored Procedure** One working stored procedure or trigger that modifies data **10 Written Report** Clear, professional prose; no bullet-point walls

50 Total (scaled to 20%)

5.5. Final Submission (40% of project)

Due: 22 June 2026, 23:59 EAT
Submission: Google Classroom (upload ZIP and/or paste GitHub link)
Checklist (minimum):

• Start-to-finish run on your laptop: install/run steps already tested • DB is populated and the app produces correct outputs for at least 3 test cases • You can explain: schema, key queries, indexing choices, and any AI components • A Git repo exists and reflects real work (not one giant last-day commit) • A fallback plan: sample inputs + expected outputs if a network/API is down **Repository must contain (minimum):**

• /schema/ — ER diagram + final DDL script

• /data/ — full dataset as CSV or JSON (or documented fetch script) • /queries/ — all SQL files labelled by purpose

• /app/ — Python application code

• /report/ — written report (max 15 pages, PDF)

• /demo/ — 5-minute screen-recorded demo video

• README.md — full setup and run instructions + AI usage disclosure **Final submission checklist (minimum):**

• One command (or exact steps) reproduces your results end-to-end

• Report includes: final schema diagram, key queries, performance results, limitations, future work

• Demo video shows: setup, core features, and at least one performance/DB feature • No secrets committed (API keys, passwords). Use .env.

Z2004: Database Management Systems IIT Madras Zanzibar

20 Technical Completeness chosen track
All minimum requirements met for

20 Code Quality Clean, commented, reproducible; Git history shows progression
20 Written Report Precise technical writing; schema diagrams; performance discussion
20 Demo Video Clear 5-minute walkthrough showing all major features **20 Innovation**
Evidence of going beyond the minimum requirements

100 Total (scaled to 40%)

6. Academic Integrity Policy

Automated plagiarism detection tools compare all submitted code across all groups. Git commit logs are evidence of individual contribution. Sharing database schemas, query logic, or Python code between groups results in a **zero for the entire team** and referral to the Academic Integrity Committee.

Allowed: discuss concepts, debugging strategies, and general approaches with other groups.

Not allowed: share artifacts (schemas, SQL files, code, datasets, notebooks, reports, slides, or videos) between groups.

Z2004 DBMS IIT Madras Zanzibar