

Project Proposal Report

1. Cover Information

Project title: GarmentIQ (Data-Driven Apparel Operations & Sales Intelligence System)

Team name: The Juls

Group Number: 47

Team members: Garv (101566625)

Course Name: System Development Project

Course Code: COMP 2154

Date: Friday 13th Feb 2026

2. Project Context and Client Focus

2.1 Project Background

The apparel industry faces challenges in managing inventory, tracking sales, and analyzing business performance in real-time. Small to medium-sized clothing brands often rely on manual processes or fragmented tools, causing inefficiencies in stock management, order fulfillment, and sales analysis. This project proposes GarmentIQ, a web-based platform that combines operational management and business intelligence to streamline decision-making and improve overall efficiency.

2.2 Client or Stakeholder Focus

Primary stakeholders include:

- Brand Owner / Manager: Needs real-time insights on inventory, sales trends, and business performance.
- Store Staff / Warehouse Staff: Requires easy data entry and update capabilities for stock, orders, and sales.
- Business Analysts / Decision Makers: Needs exportable, actionable reports to drive strategy.

Stakeholders care about accuracy, ease-of-use, and timely insights to reduce human error and improve operational decisions.

3. Problem Statement and Business Need

3.1 Business Need

Current inventory and sales tracking processes are manual or distributed across multiple platforms, leading to inefficiency, delayed reporting, and potential stockouts or overselling. Decision-making lacks data-driven support, reducing the ability to quickly respond to market demands.

3.2 Problem Statement

Clothing brand managers and staff currently lack a centralized system to manage inventory, track sales, and generate actionable insights. This results in delayed reporting, inefficient operations, and suboptimal business performance. GarmentIQ aims to solve this by providing a unified, data-driven solution.

4. Project Objectives and Scope

4.1 Project Objectives

- Develop a web-based platform for operational management of a clothing brand.
- Implement role-based access for Managers, Staff, and Analysts.
- Integrate a relational database (MySQL) to store inventory, sales, and employee activity data.
- Enable data import/export for reporting and analysis.
- Build interactive dashboards and reports using Power BI to provide actionable business insights.

4.2 Scope – In Scope

- Web interface for home page, staff login, manager login.
- Inventory management (add/update stock, sales entries).
- User roles: Manager, Staff, Analyst.
- Database design and data relations (MySQL & ERD diagrams).
- Data export to Excel and visualization in Power BI.
- Initial testing and documentation.

4.3 Scope – Out of Scope

- Development of a full-featured AI system that automatically updates, deletes, or reorders products when stock reaches minimum levels and sends email notifications to the manager or person in charge.
 - Dedicated mobile application version of the system.
-

5. Proposed Solution and System Overview

5.1 System Type and Users

GarmentIQ will be a web-based platform accessible through browsers.

- Manager: Oversees operations, updates inventory, views reports.
- Staff: Enters sales and stock data.
- BI Analyst: Accesses analytics and Power BI dashboards.

5.2 System Overview

The system will consist of:

- Front-end web interface (HTML, CSS, minimal JavaScript) for user interactions.
- Back-end database (PgAdmin4) to store, retrieve and manage data.
- Data analysis layer exporting data to Excel, then visualized in Power BI.
- Design & modeling tools (MySQL for schema designing, Visual Paradigm for UML/use cases, Draw.io for relational diagrams).

Users log in → perform tasks (inventory/sales updates) → database stores data → Excel/Power BI transforms it into actionable insights.

6. Technology Stack and Tools

- Programming Languages: HTML, CSS, minimal JavaScript.
- Frameworks/Tools: Visual Studio, ASP.NET (for web development).
- Database: PgAdmin4 & MySQL Workbench (relational database with ERD design).
- Visualization: Power BI (sales and inventory analytics).
- Modeling & Design: Visual Paradigm (UML, use cases, text analysis), Draw.io (ERD and relations).
- Version Control & Project Management: GitHub, Jira.

Justification: The stack leverages your current learning and aligns with the course goal to combine development, database design, and analytics into a single system.

7. Feasibility, Constraints, and Risks

7.1 Feasibility Summary

The project is feasible due to:

- Solo implementation with manageable scope.
- Tools and technologies already familiar from previous courses.
- Clear semester timeline aligned with course deliverables.

7.2 Constraints

- Only web-based access (no mobile app).
- Data is manually entered for initial testing.
- Fixed semester timeline and single developer.

7.3 Initial Risks and Mitigations

RISK	IMPACT	MITIGATION
DATA ENTRY ERRORS	Medium	Validation rules, error messages

DELAYED DASHBOARDS	Medium	Pre-plan Power BI model structure
SINGLE DEVELOPER	High	Focus on minimum viable features first

8. Project Plan and Schedule

8.1 Approach: Gantt Chart or Agile Backlog

Agile-style iterative development with weekly progress reports, sprints for:

- Requirement refinement → Design → Implementation → Testing → Deployment

8.2 Major Phases and Milestones

PHASE	DESCRIPTION	WEEK
REQUIREMENTS GATHERING	Collect all functional/non-functional requirements	1–2
SYSTEM DESIGN	ERD, UML diagrams, user stories	3–4
IMPLEMENTATION	Web interface, database, CRUD functions	5–8
TESTING & QA	Verify functionality, error handling	9–11
BI & REPORTING	Export to Excel, Power BI dashboards	10–12
FINAL PRESENTATION & DEMO	Slide deck, walkthrough	13–14

8.3 Task Allocation

Garv (solo): All phases, from design to implementation and reporting.

9. Team Structure and Communication Plan

- Solo project: All roles by Garv - Front-end, Back-end, Database, QA, BI, Documentation.
 - Communication via GitHub issues, self-checks, weekly progress report updates.
 - Decision-making: All decisions by developer, maintain documentation of changes.
-

10. Assumptions, Dependencies, and Success Criteria

10.1 Assumptions

- Test/dummy data is available or can be generated.
- Access to MySQL Workbench, Visual Paradigm, Power BI.
- Users behave according to defined roles.

10.2 Dependencies

- MySQL and web hosting environment availability.
- To Operate Power BI on MacBook, I need Parallel Subscriptions (Already Bought).

10.3 Success Criteria

- Functional web-based system with role-based access.
 - Database integration with accurate CRUD operations.
 - BI dashboards display correct inventory/sales insights.
 - Completion of documentation, diagrams, and project report.
-

11. Summary and Next Steps

GarmentIQ provides a centralized, data-driven solution for clothing brand operations and sales intelligence. The project addresses current inefficiencies in inventory management, sales tracking, and reporting.

Next steps:

- Refine architecture and design using Visual Paradigm & Draw.io.
 - Begin database creation in MySQL, populate with data(real/dummy).
 - Start web interface development.
 - Track weekly progress and prepare for Power BI reporting pipeline.
-
-