AI USAGE LOG & REFLECTION REPORT - PHASE 2

SCHOOL EQUIPMENT LENDING PORTAL

SE ZG503 – FULL STACK APPLICATION DEVELOPMENT

Author: Sarumathy G.
Roll No: 2024TM93153
Date: 09 November 2025

1. OVERVIEW

The SCHOOL EQUIPMENT LENDING PORTAL is a full-stack MERN (MongoDB, Express, React, Node.js) application that allows students and staff to borrow, return, and manage school equipment efficiently.

The portal supports user authentication, item booking, return tracking, and an admin panel for monitoring lending activity.

For this phase, I used AI tools **only as a supportive guide**, to review ideas, clarify doubts, and refine specific sections of syntax or documentation. All major coding, logic design, database modelling, and component development were **conceptualized and implemented entirely by me**.

2. AI USAGE LOG

No.	Area	AI Tool	Type of Support	My Own Work
1	**Initial Planning**	ChatGPT	Used to discuss the structure of a MERN stack app and verify best practices in folder organization.	Created my own backend and frontend structure manually.
2	**Schema Validation**	ChatGPT	Referred briefly to Joi validation examples to check correct syntax for optional fields.	Wrote the complete schema definitions for User, Item, and Booking models myself.
3	**Auth Token Refinement**	ChatGPT	Reviewed my JWT token generation logic to ensure secure	Implemented token creation, middleware, and

No.	Area	AI Tool	Type of Support	My Own Work
			signing and claims inclusion.	error handling fully on my own.
4	**Swagger Documentation Format**	ChatGPT	Looked up sample syntax to understand annotation style for route documentation.	Documented all routes manually and tested Swagger UI locally.
5	**UI Enhancement Ideas**	ChatGPT	Asked for general ideas on improving navbar interactions and modal confirmation UX.	Designed, styled, and coded the components myself using React + Tailwind.
6	**Error Troubleshooting**	ChatGPT	Used to interpret a few runtime messages (like import/export or token verification issues).	Debugged, tested, and corrected all errors independently.
7	**README & Report Review**	ChatGPT	Used for language tone polishing and formatting structure for README and this reflection.	Authored entire content and diagrams personally.

3. REFLECTION ON AI USE

3.1 AI AS A REVIEWER, NOT A BUILDER

AI was primarily used as a **reviewer** rather than a code generator. For instance, when I implemented the jwt.sign() logic for authentication, I had already written the function, and AI was used only to confirm the correct claim structure and security best practices. Similarly, while setting up routes, I used AI to double-check naming conventions but coded and tested everything myself.

This ensured that my project remained **fully original and personally built**, with AI acting as a quick feedback loop rather than a dependency.

3.2 ENHANCING CONFIDENCE IN SYNTAX AND STANDARDS

AI helped confirm my understanding of tools like Joi, Swagger, and Express middleware's. It didn't introduce new logic or structures but gave **confidence that what I was doing aligned with professional syntax standards**.

This allowed me to focus more on clean code practices and logical consistency.

3.3 ACCELERATING CLARITY, NOT DEVELOPMENT

Whenever I faced doubt on the correct use of optional parameters in Joi or role-based middleware design, I referred to AI for clarification. By asking short, focused questions, I was able to resolve conceptual blockers faster without losing ownership of the coding process.

This made AI an **efficiency enhancer** rather than a productivity crutch.

3.4 LEARNING REINFORCEMENT THROUGH VERIFICATION

AI helped me **verify** solutions I had already designed. For example, when I completed the CRUD routes for Items and Bookings, I asked AI to "review if this follows REST best practices."

The feedback reinforced my learning and improved naming clarity (like using /api/items/:id instead of /api/item/get).

Such iterative feedback improved design precision while maintaining self-driven logic.

3.5 DOCUMENTATION AND PRESENTATION POLISHING

I used AI to help improve the structure of written sections like this reflection and README. However, the actual descriptions, architecture diagrams, and API flow explanations were all derived from my hands-on understanding of the system. AI only assisted with grammar refinement and formatting consistency.

4. KEY LEARNINGS

Area	My Takeaway
Backend Logic	Deepened understanding of Express middleware flow, JWT token handling, and modular route design.
Database Design	Understood how to structure linked collections for User, Item, and Booking using Mongoose.
Swagger/OpenAPI	Learned to write inline route annotations manually for automatic documentation.
Frontend Logic	Improved confidence in managing React component hierarchy, state, and context for authentication.
AI Ethics	Learned how to use AI responsibly, for reviewing, verifying, and learning, not for copying or delegation.

5. RESPONSIBLE AI PRACTICE

Throughout the project, I made sure:

- **No part of the main source code was auto-generated.**
- All integrations (MongoDB Atlas, JWT, Swagger, React UI) were manually configured.
- Every AI suggestion was **reviewed, rewritten, and tested by me** before including in code.
- I could explain every line of logic, model, and route without referring to AI outputs.

This ensured both **academic integrity** and genuine skill development.

6. CONCLUSION

AI acted as a **learning assistant** that improved confidence, not as a development shortcut. It supported me in understanding edge cases, verifying best practices, and polishing written reports, while the complete system, from routes to UI, was **entirely conceptualized, built, and debugged by me**. Using AI responsibly taught me how to balance self-reliance with intelligent support, a vital skill for real-world full-stack engineering.