



AI Use Documentation - Individual Final Project

CS341 – Web Technologies (Spring 2025)

Project Name: Cr8Kit (Creative Equipment Rental Platform)

Name: Jerome Selorm Kwame Adedze

ID: 66662027

Date: December 17, 2025

Introduction

This document provides a clear and structured disclosure of Artificial Intelligence tools used in the development of the Cr8Kit web application, in compliance with course requirements. AI tools were used as development assistants for code structure, debugging, architectural guidance, and UI refinement. Approximately 45% of the total codebase was AI-assisted, with all outputs reviewed, modified, and validated by my individual effort.

1. AI Tools Used

Dates Used: December 7 – December 17, 2025

- i) Antigravity (Google DeepMind)
- ii) Cursor AI – Anthropic (Claude Sonnet 4.5)
- iii) ChatGPT – OpenAI (GPT-4)

2. Purpose and Scope of AI Usage

AI tools were primarily used to:

- Generating boilerplate frontend and backend code
- Assisting with Supabase integration and database security
- Implementing complex JavaScript logic (async flows, DOM updates)
- Aid in designing clean and responsive CSS layouts
- Refactoring project structure for scalability

AI was not used to replace overall decision-making or testing. All architectural decisions, specific context customizations, and validation were performed manually.

3. Prompts Used

The following prompts represent how AI tools were consulted during the development of the Cr8Kit project. The prompts were used in an assistive and exploratory manner, mainly to clarify implementation approaches, resolve confusing bugs, or generate initial drafts that were subsequently refined, corrected, and integrated manually.

Prompts

1. "Suggest improvements to the layout of the equipment section so that items are displayed more clearly on wider screens."
2. "Review my project rubric and highlight any missing technical or documentation requirements."
3. "Explain why my users and equipment tabs may not be displaying data and suggest possible fixes."
4. "Provide guidance on aligning Row Level Security policies when authentication identifiers differ ."
5. "Help me implement a Ghana Card verification workflow, including image uploads and an admin approval dashboard."
6. "Remove hardcoded UI numbers and price sliders on the browse page and replace them with dynamic filtering."
7. "Help me create a functional rating and review system that triggers after a user marks equipment as returned."
8. "Fix broken sign-out functionality and implement a hover-based profile dropdown menu across all pages."
9. "Enable multi-image selection for Cloudinary uploads instead of uploading files one by one."
10. "Fix the navigation logic so clicking an item in the browse list leads to its specific details page rather than a hardcoded placeholder."
11. "Fix the navigation logic in the favorites section to ensure users are redirected to the correct equipment details page."
12. "Improve the category filtering logic in the browse section to prevent mismatched results."
13. "Explain the technical challenges of integrating a Supabase backend with a static HTML/JS frontend for an academic project."
14. "Resolve the git checkout error and confirm the repository is in the correct detached HEAD state for the requested commit."
15. "Fix the sign-out button in the profile header, which is currently unresponsive during user sessions."
16. "Fix the my equipment section, make it wider to display the equipment well"
17. "Analyze my project rubric and tell me what is lacking"
18. "Add ghana_card_image variable to the database to actually verify users"

4. Outputs Generated and Used

Frontend

- **Responsive CSS Layouts:** Generated CSS Grid layouts for the "My Equipment" section to support wider screens and responsive equipment cards () .
- **Verification Workflows:** Created interactive modals for Ghana Card submission, including format validation for Card IDs and photo upload integration () .
- **DOM Manipulation:** Implemented logic for dynamic notification badges, hover-activated profile menus, and "Mark as Returned" button states () .
- **Dynamic Calendar:** Developed a custom JavaScript calendar that replaces hardcoded dates with real-time monthly views and availability checking () .
- **Navigation & Routing Fixes:** Implemented dynamic URL parameter handling to ensure that clicking on "Favorites" or "Browse" items correctly routes the user to the specific equipment-details.html?id=[ID] page.
- **Sign-Out Logic:** Refactored the authentication logout handler to properly clear local storage and session tokens, resolving the unresponsive button issue in the profile header.
- **Category Filter Refinement:** Updated the JavaScript filtering logic to use exact string matching for equipment categories, ensuring "Cameras" does not incorrectly return "Drones."
- Responsive CSS Grid layouts for equipment cards (css/profile.css)
- JavaScript logic for dashboards and verification workflows
- DOM manipulation for admin actions (verify, delete, approve)

Backend & Database

- Supabase Row Level Security (RLS) policies (database/policies.sql)
- SQL schema extensions (Ghana Card ID & image fields)
- Asynchronous data fetching and mutation logic
- Asynchronous CRUD Logic: Built robust JavaScript functions to handle INSERT, UPDATE, and DELETE operations via the Supabase client, including complex multi-table joins for bookings and reviews () .
- Security Hardening: Implemented auth.jwt() ->> 'email' claims within database policies to verify user identity without accessing protected internal schemas () .

5. Complex Technical Areas Where AI Was Critical

a) Row Level Security (RLS)

Implemented database-level access control to ensure users can only read or modify their own data. Policies required non-trivial SQL subqueries and identity mapping.

b) Supabase Identity Mapping

Resolved the mismatch between Supabase Auth UUIDs and application Integer IDs by matching users via unique email addresses.

c) Real-Time Image Handling

Integrated Cloudinary for Ghana Card uploads, including async handling, loading states, and secure URL storage.

d) Sophisticated Data Analytics and Aggregation

Replacing hardcoded dashboard numbers required complex database aggregation that the AI helped write, Statistical Queries: Writing SQL and JavaScript logic to calculate Total Spent, Pending Requests, and Active Rentals by aggregating data across the bookings and equipment tables.

e) Automated Status and State Transitions

AI was critical in managing the "lifecycle" of various entities within the app, moving beyond simple data entry into complex logic:

- **Booking State Machine:** Implementing logic that automatically transitions a booking from Pending to Confirmed to Returned, while simultaneously updating the equipment's Availability status in the database.
- **Verification Lifecycle:** Managing user transitions from Unverified to Pending to Verified (or Rejected), including automated in-app notifications and locking/unlocking features (like the "List Item" page) based on these statuses.

6. Post-Processing and Verification

All AI-generated outputs were:

- Refactored to match and align with the project's branding (in terms of color themes and design)
- Validated for security and correctness

Testing included:

- Attempted unauthorized access (RLS validation)
- CRUD flow testing
- Responsive testing (desktop version)

8. Reflection

AI significantly accelerated development and learning, reducing the project timeline by over 60%. However, meaningful outcomes were achieved only through human judgment, testing, and contextual adaptation to Ghana-specific requirements. This experience streamlined and sped up my learning process in full-stack development. With the help of AI, I have gained practical knowledge in connecting frontend interfaces to backend services, managing data flow between the client and server, and integrating authentication and database logic into a cohesive web application.