

# AI Development Log (Week 1)

Date: 2026-02-16

Project: CollabBoard MVP-1

## 1) Tools and Workflow

- AI tools used: Codex (primary implementation/review), Cursor (paired implementation), Claude (adversarial requirement checks).
- Workflow:
  1. Parse rubric + requirement PDF into implementation contracts.
  2. Convert contracts into PRESEARCH.md, PRD.md, MVP.md, DECISIONS.md, TASKS.md.
  3. Implement multiplayer core first (auth, presence, object sync), then AI backend.
  4. Run lint/build/e2e checks and iterate on blockers.
  5. Deploy to Firebase Hosting + Functions and verify with live smoke tests.

## 2) MCP Usage

- Linear MCP was used to map work into tickets and maintain execution order (planning + delivery tracking).
- Result: reduced ad-hoc task switching and improved traceability from requirements to implementation.

## 3) Effective Prompts (Actual)

1. "review the code and requirements do /review and commit"
2. "can you check those and see why we didnt do those? [requirements coverage feedback]"
3. "ok lets work on mvp and publish it. let me know what you need ahead of time. each mvp might have .env. make sure not push it"
4. "there was issue after moving the card the user's window moved. it should stay"
5. "include all of the changes"

## 4) Code Analysis (AI vs Hand-Written)

- Estimated split for Week 1:
  - AI-generated first draft code/docs: 70%
  - Human-directed edits/refinement/validation: 30%
- Higher human input areas:
  - Requirement interpretation and scope decisions.
  - Release/commit boundaries and production verification.

## 5) Strengths and Limitations

- Strengths:
  - Fast conversion of rubric text into executable plans and contracts.

- Rapid bug triage/fixes (drag viewport regression, auth/CORS behavior).
- Consistent cross-file updates (code + docs + deployment config).
- Limitations:
  - Requires strict review for concurrency semantics and security edges.
  - Needs human guardrails for commit hygiene and artifact inclusion.
  - E2E auth flows still need manual/secure session setup.

## 6) Key Learnings

- Define schemas + conflict semantics early; it prevents downstream ambiguity.
- Server-side idempotency and queue semantics must be explicit for multi-user AI.
- Realtime UX bugs are often event-propagation/state-sync interactions, not isolated component bugs.
- "Deploy + smoke verify" should be part of Done criteria, not a postscript.