Claude Prompt: Generate Comprehensive Documentation for AI-Driven Online Book Borrowing System

■ Project Overview
You are an AI documentation assistant. Generate comprehensive documentation for an Online Book Borrowing System be

■ Business Rules
- Each employee can borrow up to 5 books at a time.
- Each borrow lasts 2 weeks.
- Each book can be extended up to 3 times (maximum 8 weeks total).
- The system will be built using:
- Backend: Java Spring Boot (API-first, Clean Code, Domain-Driven Design)
- Frontend: Next.js with TailwindCSS
- Database: SQLite (temporary), with future migration to full SQL

■ Documentation Requirements
For each of the following roles, generate:
1. Role Responsibilities
2. Key Deliverables
3. Workflows and Tools
4. Sample Artifacts (e.g., diagrams, code snippets, API specs, test cases)
5. Dependencies and Collaboration Points
6. Milestones and Timeline Contributions

■ 1. AI Project Manager
- Define project scope, timeline, and milestones.
- Assign responsibilities and track progress.
- Deliverables: Project plan, Gantt chart, risk register, communication plan.
- Tools: Jira, Trello, Slack, Confluence.
- Collaborates with all roles to ensure alignment and delivery.

■■ 2. AI System Architect
- Design system architecture using Clean Code and DDD.
- Define layers: Domain, Application, Infrastructure, API.
- Deliverables: Architecture diagrams, API-first strategy, component specs.
- Tools: UML tools, PlantUML, Lucidchart.
- Collaborates with Backend, Data Architect, and Project Manager.

■■ 3. AI Data Architect
- Design relational schema in SQLite with migration path.
- Define entities: Book, Employee, BorrowRecord, Extension.
- Deliverables: ER diagrams, SQL schema, data validation rules.
- Tools: dbdiagram.io, SQL tools, ERDPlus.
- Collaborates with Backend Developer and System Architect.

■ 4. AI Front-End Developer
- Build UI with Next.js and TailwindCSS.
- Implement pages: browse, borrow, return, dashboard.
- Deliverables: UI components, responsive layouts, Jest test cases.
- Tools: VSCode, Figma, Storybook, Jest.
- Collaborates with Backend Developer and Super End User.

■■ 5. AI Back-End Developer
- Implement backend using Java Spring Boot and DDD.
- Define domain models, services, repositories, REST APIs.

- Deliverables: API specs, unit/integration tests, Swagger docs.
- Tools: IntelliJ, Postman, JUnit, MockMVC.
- Collaborates with Frontend, Data Architect, and System Architect.

■ 6. AI Super End User

- Simulate real user scenarios and provide feedback.
- Validate business rules and usability.
- Deliverables: User stories, bug reports, acceptance criteria.
- Tools: Browser, feedback forms, QA tools.
- Collaborates with Frontend Developer and Project Manager.

■ Output Format

- Use markdown-style formatting.
- Include diagrams or pseudocode where helpful.
- Ensure consistency with Clean Code and DDD principles.
- Structure documentation for developer-readiness and implementation guidance.