# **Project Weaver — AI-Powered Project Management Platform**

**Date:** September 16, 2025 • **Prepared by:** Development Team

## **1) Introduction**

Weaver is an AI-first platform for modern project management and software architecture planning. It pairs Google Gemini–powered assistance with intuitive visuals (Mermaid.js) to accelerate ideation → design → delivery for technical and non-technical teams.

## **2) Objectives**

**Primary**

1. MVP to Production: Build a cloud-based system that can grow from a basic version to a full multi-user production system.
2. AI-First: Use Gemini to help with planning, architecture, and smart assistance.
3. Scalability: Design the system to expand easily and keep different users' data separate.

## **3) Scope**

1. Foundational architecture; Auth (Google OAuth 2.0, JWT), roles/permissions
2. AI service (Gemini, prompt orchestration, caching, guardrails)
3. Project/collab (CRUD, context, chat, presence, activity)
4. Diagramming (Mermaid.js, versioning/exports)
5. Data layer (DynamoDB + Redis), API gateway (Nginx), containers (Docker), orchestration (Kubernetes)
6. Observability (Prometheus, Grafana), CI/CD, testing, OpenAPI docs

## **4) Architecture Overview**

## 

## **5) Technology Stack**

**Frontend:** Next.js 15 + React 19, Tailwind, React hooks/context, server actions, custom UI.  
**Backend:** [Next.js](http://next.js) 15 +, Google OAuth 2.0 + JWT, Gemini, DynamoDB, Redis.  
**Infra/Tooling:** Nginx, Docker, Kubernetes, (Mesh later), Prometheus/Grafana, ELK, GitHub + Actions, ESLint/Prettier, Jest/RTL, OpenAPI.

## **6) 12-Week Implementation Plan (Sep 16 → Dec 8, 2025)**

**Weeks 1–3: Foundation**

* Cloud environments, baseline IaC; repo + monorepo scaffolding & package standards
* CI/CD (build/test/deploy), base Docker images
* API Gateway skeleton & routes; skeleton services; auth wiring (Google)

**Weeks 4–6: Core Platform**

* Auth service (Google OAuth, JWT), logging
* Project service (CRUD, tenancy); DynamoDB schema + seed tooling
* Nginx policies (auth, rate limits); initial frontend (auth, projects list/detail)

**Weeks 7–9: AI Integration**

* AI service (Gemini) with prompt templates, persona/guardrails
* Redis caching, retries, circuit breakers; error handling/observability for AI calls
* UX: AI-assisted planning and Q&A flows in projects

**Weeks 10–12: Collaboration, Diagrams, Prod Readiness**

* Real-time chat/presence, context threads; activity feeds
* Mermaid.js diagram service (render, version, export); file export/upload flows
* Alerting and Monitoring
* **Exit criteria**: MVP complete and readiness by end of Week 12

## **7) Risks & Mitigations**

* **AI integration** → Start small with AI, test it well, and have backup plans.
* **Service boundaries** → Build as one piece initially, then break off parts as they become clear; limit how much services talk to each other.
* **DynamoDB unknowns** → Design the data structure first, test it with fake data, and use other database types if needed.
* **Scope & schedule** → Keep the task list clean, have checkpoints, show progress every two weeks, and focus on what's most important.