## Aegis Platform Implementation Plan

This plan is divided into four distinct phases, allowing for iterative development, testing, and feedback.

- **Phase 1: Minimum Viable Product (MVP)** - Build the core, end-to-end risk management loop.
- **Phase 2: Core Enhancements & Integrations** - Add key integrations and improve the user experience.
- **Phase 3: Intelligence Layer** - Weave in the LLM-powered automation features.
- **Phase 4: Enterprise Readiness** - Add features for scale, auditing, and advanced reporting.

## Phase 1: MVP - The Core Risk Management Loop

The goal of this phase is to have a functional, end-to-end platform that allows a user to perform a basic risk assessment and track it to completion.

### Epic 1: Foundation & Project Setup

- ☐ **Task:** Initialize Git repository and establish branching strategy (e.g., GitFlow).
- ☐ **Task:** Set up cloud infrastructure (e.g., Azure App Service, Vercel, AWS Amplify).
- ☐ **Task:** Configure CI/CD pipeline for automated testing and deployment.
- ☐ **Task:** Set up project boilerplate for Frontend (Next.js) and Backend (FastAPI/Node.js).
- ☐ **Task:** Create initial database schema in PostgreSQL.
- ☐ **Task:** Implement basic design system tokens (colors, fonts, spacing) in the frontend.

### Epic 2: User Authentication & Roles (Core)

- ☐ **Task:** Integrate Microsoft Entra ID using OAuth 2.0 for user login/logout.
- ☐ **Task:** Create database tables for Users and basic Roles (Admin, Analyst, ReadOnly).
- ☐ **Task:** Implement backend middleware to protect authenticated routes.
- ☐ **Task:** Build the frontend login page and logic for handling authentication tokens.
- ☐ **Task:** Create a simple user profile page.

### Epic 3: Asset Management (Core)

- ☐ **Task:** Design and implement the `Assets` table in the database.
- ☐ **Task:** Build backend API endpoints (CRUD - Create, Read, Update, Delete) for assets.
- ☐ **Task:** Create frontend UI to list, add, edit, and view assets.
- ☐ **Task:** Implement a basic CSV import function for assets.

### Epic 4: Assessment & Analysis (Core)

- ☐ **Task:** Create a database seeding script to load NIST CSF and CIS Controls frameworks.
- ☐ **Task:** Build UI to start a new assessment by selecting assets and a framework.
- ☐ **Task:** Develop the main assessment interface where users can set a status for each control (Implemented, Not Implemented, etc.).

- ☐ **Task:** Implement the logic to automatically create a "Risk" in the Risk Register when a control is marked "Not Implemented."
- ☐ **Task:** Build the basic Risk Register UI to view a list of identified risks.
- ☐ **Task:** Implement the configurable Likelihood/Impact matrix and allow users to score risks.

**Epic 5: Mitigation & Reporting (Core)**

- ☐ **Task:** From the Risk Register, allow users to create a simple remediation task (Title, Assignee, Status).
- ☐ **Task:** Build a basic Kanban or list view to track mitigation task status.
- ☐ **Task:** Create a single, simple report template (e.g., PDF export of the Risk Register).

**Epic 6: Basic Dashboard**

- ☐ **Task:** Design a simple, single dashboard for all users.
- ☐ **Task:** Add widgets for basic counts: Total Assets, Open Risks, Assessments in Progress.

---

## Phase 2: Core Enhancements & Integrations

With the core loop functional, this phase focuses on adding the powerful integrations and improving the UX.

**Epic 7: Advanced UX/UI Polish**

- ☐ **Task:** Implement the gradient accent colors and other visual design elements.
- ☐ **Task:** Add smooth page transitions and micro-animations on interactive elements.
- ☐ **Task:** Implement skeleton loaders for a better data-loading experience.
- ☐ **Task:** Implement both Light and Dark modes.

**Epic 8: Evidence & Documentation**

- ☐ **Task:** Implement secure file upload capability for evidence (link to controls and tasks).
- ☐ **Task:** Add rich-text editor for control narratives and comments.
- ☐ **Task:** Build the "Awaiting Review" workflow gate for mitigation tasks.

**Epic 9: OpenVAS Integration**

- ☐ **Task:** Develop a backend service to connect to the OpenVAS API.
- ☐ **Task:** Create a UI for admins to configure the OpenVAS connection details securely.
- ☐ **Task:** Implement the logic to correlate OpenVAS vulnerabilities with Aegis assets.
- ☐ **Task:** Create a workflow to automatically flag controls (CIS 7) or create risks based on high-severity findings.

**Epic 10: OpenCTI Integration**

- ☐ **Task:** Develop a backend service to connect to the OpenCTI GraphQL API.
- ☐ **Task:** On the Risk Register page, create a new UI tab or section called "Threat Intel."
- ☐ **Task:** When a risk is viewed, query OpenCTI for related threat actors, malware, or campaigns and display the results.

**Epic 11: Role-Based Dashboards**

- □ **Task:** Refactor the basic dashboard into a dynamic, widget-based system.
- □ **Task:** Design and implement the CISO Cockpit, Analyst Workbench, and System Owner Inbox views.

## Phase 3: Intelligence Layer (LLM Features)

This phase introduces the AI capabilities that differentiate the platform.

**Epic 12: LLM Service Integration**

- □ **Task:** Select an LLM provider (e.g., Azure OpenAI) and set up secure API access.
- □ **Task:** Create a centralized backend service for managing prompts and parsing LLM responses.

**Epic 13: AI-Powered Features**

- □ **Task: (Evidence Analysis):** Add a button to "Analyze Evidence." Wire it to the LLM service to scan uploaded documents and provide a summary.
- □ **Task: (Narrative Generation):** Add a "Suggest Narrative" button that uses evidence summaries to generate a draft narrative for a control.
- □ **Task: (Risk Statement Generation):** Add a "Generate with AI" button on the Risk Register that uses technical data and threat intel to draft a formal risk statement.
- □ **Task: (Remediation Suggestions):** Add a "Suggest Plan" button on mitigation tasks to generate a high-level, step-by-step remediation plan.
- □ **Task: (Executive Summary):** In the reporting module, add a "Generate Executive Summary" button that analyzes dashboard metrics to produce a narrative for leadership.

## Phase 4: Enterprise Readiness

This final phase adds features required for large-scale adoption, security, and compliance.

**Epic 14: Advanced Reporting & Auditing**

- □ **Task:** Develop the custom report builder with a drag-and-drop interface.
- □ **Task:** Implement report scheduling and email distribution.
- □ **Task:** Create a comprehensive audit trail for every significant action in the platform.
- □ **Task:** Build a UI to view and search the audit logs.

**Epic 15: Advanced RBAC**

- □ **Task:** Extend the roles system to allow for custom, user-defined roles.
- □ **Task:** Implement fine-grained permissions (e.g., restricting access to assets or modules based on role).

**Epic 16: Documentation**

- □ **Task:** Set up the Astro project for documentation.
- □ **Task:** Write comprehensive user guides for all major features and personas.
- □ **Task:** Create admin guides for setting up integrations and authentication.

**Epic 17: Scalability & Security Hardening**

- □ **Task:** Conduct performance and load testing on the platform.
- □ **Task:** Perform a third-party penetration test and remediate findings.
- □ **Task:** Optimize database queries and add caching where appropriate.