Background & Problem Statement

Students today often rely on multiple Learning Management Systems (LMS) such as Google Classroom, Canvas, and Moodle. This fragmentation creates inefficiency and cognitive load, requiring students to manage multiple logins and check different platforms to track courses, assignments, and grades. Most LMS platforms lack advanced features such as GPA forecasting, personalized goal tracking, and AI-driven tutoring support. ClassPilot aims to solve these challenges by serving as an AI-powered student operating system that unifies multiple LMSs into a single interface, providing clarity, insights, planning tools, and intelligent assistance.

Objectives & Goals

The objective of ClassPilot is to create a unified academic management platform that empowers students to manage their coursework effectively and securely. Goals include: consolidating course and assignment data into one dashboard, enabling secure LMS integrations with read/write functionality where possible, offering GPA tracking and simulations, simplifying submission flows with deep links, scaffolding AI tutor features for future study support, and ensuring that the system is modular, scalable, and secure.

Scope

The MVP scope consists of: a Next.js web application, authentication and session management using Clerk with Google OAuth, a unified dashboard displaying courses, assignments, and deadlines, integration with Google Classroom (read-only plus deep-link submissions), Canvas (read/write functionality), Moodle (read-only or locked fallback), a GPA engine for calculations and simulations, observability with logging, error tracking, and monitoring, and containerized infrastructure using Docker Compose and Postgres. Out-of-scope features for MVP include mobile applications, gamification, note ingestion, advanced AI tutoring beyond stubs, and payment or subscription flows.

User Stories

1. As a student, I want to log in with Google so my courses automatically sync. 2. As a student, I want to see all my assignments in one place so I don’t miss deadlines. 3. As a student, I want to calculate my GPA and simulate outcomes so I can track academic progress. 4. As a student, I want to submit work via deep links into Google Classroom so I don’t have to switch platforms. 5. As a student, I want my data to remain secure so I can trust the platform with sensitive academic information.

Functional Requirements

Authentication & Sessions: must support Google OAuth and Clerk-managed sessions, with encrypted token handling and secure cookie storage. Course & Assignment Management: fetch and unify course and assignment data across LMSs, display upcoming deadlines and overdue tasks, support assignment submission to Canvas with deep links for Google Classroom. GPA Engine: calculate GPA accurately based on assignment and course data, allow simulations for potential outcomes. AI Tutor Stub: provide a placeholder question-answering interface for future RAG-based features. Infrastructure: Postgres database managed through Prisma ORM, Docker Compose setup for local development, and CI/CD workflows to ensure quality and deployment readiness.

Non-Functional Requirements

Security: enforce strict input validation, implement RBAC, and apply row-level security in the database. Performance: dashboard should load within two seconds at the 95th percentile under normal usage. Scalability: must handle at least 10,000 concurrent users without service degradation. Maintainability: follow modular Nx monorepo structure, enforce strict TypeScript, and provide documentation for all packages. Observability: implement structured logging (pino), metrics and tracing with OpenTelemetry, and error tracking with Sentry.

Success Metrics

Adoption: at least 500 students onboarded during MVP pilot phase. Retention: at least 60% weekly active users one month post-launch. Accuracy: GPA engine must achieve ≥99% accuracy compared to institutional GPA formulas. Uptime: service reliability of ≥99.5% during semester. Security: no unresolved critical or high vulnerabilities in production dependencies.

Risks & Assumptions

Risks: Google Classroom and Moodle may restrict write access; students may hesitate to connect multiple LMS accounts due to privacy concerns. Assumptions: users will provide valid OAuth credentials and Canvas API tokens; pilot schools will permit API access during testing.