# Appendix

## Glossary

* **RLS** - Row-Level Security (database-level access control)
* **RSC** - React Server Components
* **BaaS** - Backend as a Service
* **JSONB** - PostgreSQL’s binary JSON data type
* **SWOT** - Strengths, Weaknesses, Opportunities, Threats analysis
* **KPI** - Key Performance Indicator
* **FY** - Fiscal Year

## Key Dependencies Documentation

* [Next.js 14 Docs](https://nextjs.org/docs)
* [Supabase Docs](https://supabase.com/docs)
* [React Hook Form](https://react-hook-form.com/)
* [Zod](https://zod.dev/)
* [TailwindCSS](https://tailwindcss.com/)
* [shadcn/ui](https://ui.shadcn.com/)

## Architecture Decision Records (ADRs)

**ADR-001: Hybrid Data Model** - **Decision**: Use normalized relational tables + JSONB - **Rationale**: Balance queryability with flexibility - **Status**: Approved

**ADR-002: Server Components First** - **Decision**: Default to Server Components, minimize client JS - **Rationale**: Better performance, SEO, security - **Status**: Approved

**ADR-003: Supabase RLS for Security** - **Decision**: Enforce all access control via RLS policies - **Rationale**: Defense in depth, prevents bypass - **Status**: Approved

**ADR-004: No Custom API Layer** - **Decision**: Use Supabase PostgREST, no custom REST/GraphQL - **Rationale**: Reduce complexity, leverage auto-generated API - **Status**: Approved

**Document Status:** ✅ Complete **Last Reviewed:** January 9, 2025 **Next Review:** February 9, 2025