# Prism - Modern Project Management Platform

## Table of Contents

1. [Introduction](#introduction)
2. [Existing Problems](#existing-problems)
   * [The Complexity Crisis](#the-complexity-crisis)
   * [The Small Team Problem](#the-small-team-problem)
   * [Why Simple Project Management Matters](#why-simple-project-management-matters)
   * [The Student and Startup Reality](#the-student-and-startup-reality)
   * [Research-Backed Evidence](#research-backed-evidence)
   * [The Gap in the Market](#the-gap-in-the-market)
   * [The Solution Approach](#the-solution-approach)
3. [Objectives](#objectives)
4. [Requirements](#requirements)
   * [Functional Requirements](#functional-requirements)
   * [Non-Functional Requirements](#non-functional-requirements)
5. [Design](#design)
   * [User Flow](#user-flow)
   * [Database Design](#database-design)
   * [Screen Design](#screen-design)
   * [API Design](#api-design)
6. [Tech Stack and Implementation](#tech-stack-and-implementation)
7. [Deployment](#deployment)
8. [Conclusion](#conclusion)

## Introduction

Prism is a modern project management platform designed to help teams organize and track their work efficiently. It provides a sleek, feature-rich environment for organization-based project management, sprint planning, and intuitive issue tracking with a beautiful, animated UI. The application is built using Next.js 15 and incorporates modern web development practices to solve the core problems that plague traditional project management tools.

## Existing Problems

The project management software landscape is filled with tools that have lost sight of their primary purpose: helping teams get work done efficiently. Through extensive research and analysis, several critical problems have been identified with traditional project management tools that particularly affect small teams, startups, and students.

### The Complexity Crisis

With so many features, the interface can feel cluttered, and it takes time to fully grasp - this is a common complaint about popular project management tools. Project management tools are often complicated and overwhelming, requiring users to deal with multiple features and functions that may not be relevant or useful for their projects. This complexity creates several issues:

**Feature Overload**: Traditional tools try to be everything to everyone, resulting in interfaces packed with buttons, menus, and options that confuse rather than help. As noted in industry analysis, [tool complexity must match team sophistication](https://complex.so/insights/small-team-vs-micro-team-project-management), yet many tools ignore this principle.

**Steep Learning Curves**: [Research from MIT Sloan on technology adoption](https://complex.so/insights/small-team-vs-micro-team-project-management) shows that micro teams want to start using tools immediately. They don’t have time for training sessions or complex setups. This creates a significant barrier for students and small startups who need to be productive from day one.

### The Small Team Problem

Small teams and startups face unique challenges that traditional project management tools fail to address:

**Mismatched Tool Sophistication**: [Tool complexity must match team sophistication](https://complex.so/insights/small-team-vs-micro-team-project-management). Most traditional tools are designed for large enterprises with dedicated project managers and extensive training budgets. As research shows, these tools [may lack functionality required for small teams while being overly complex](https://springmanconsulting.com/detailed-review-of-basecamp-project-management-tool/).

**Excessive Costs**: Many tools have pricing models that don’t work for small teams. [Current market analysis](https://startup.unitelvoice.com/best-project-management-tools) shows that even basic plans of popular tools can be expensive for student projects or early-stage startups.

**Over-Engineering for Simple Needs**: If your projects are relatively straightforward and your team is small, you don’t need enterprise-level complexity. [Industry reviews](https://funficient.medium.com/unpacking-the-power-and-pitfalls-of-project-management-with-basecamp-63a783364fa2) consistently highlight that simple, intuitive tools work better for teams who value speed and clarity over complexity.

### Why Simple Project Management Matters

Project management tools are essential for any team, regardless of size, because they:

**Prevent Work from Falling Through Cracks**: Without proper tracking, tasks get forgotten, deadlines are missed, and team members duplicate effort.

**Enable Clear Communication**: Teams need a central place to discuss tasks, share updates, and track progress without endless email chains or chat messages.

**Provide Visibility**: Everyone should know what others are working on, what’s coming next, and whether the project is on track.

**Support Growth**: As teams grow from 2-3 people to 10-15 people, they need systems that can scale with them without requiring a complete tool change.

### The Student and Startup Reality

For students working on group projects and small startups, the problems are even more pronounced:

**Limited Budgets**: Students can’t afford expensive enterprise tools, and startups need to watch every dollar. [Current pricing analysis](https://thedigitalprojectmanager.com/tools/best-project-management-software/) shows significant cost barriers for small teams.

**Time Constraints**: [Teams need tools that are intuitive and require minimal setup time](https://complex.so/insights/small-team-vs-micro-team-project-management) because there’s no time for extensive training or onboarding processes.

**Changing Team Composition**: Student teams and startup teams change frequently, so tools need to be intuitive enough that new members can contribute immediately without extensive training.

**Focus on Core Work**: The tool should help teams focus on their actual project work, not on learning how to use the project management software. As [industry analysis shows](https://funficient.medium.com/unpacking-the-power-and-pitfalls-of-project-management-with-basecamp-63a783364fa2), users shouldn’t have to figure out where to find information.

### Research-Backed Evidence

Multiple studies and industry reports support these findings:

**Academic Research:** - MIT Sloan research on technology adoption shows that micro teams want to start using tools immediately without training sessions or complex setups ([Complex.so - Small Team vs Micro Team Project Management](https://complex.so/insights/small-team-vs-micro-team-project-management)) - PMI whitepaper on complexity management reveals significant challenges in bridging complexity concepts with managerial reality ([PMI - Complexity Management for Projects](https://www.pmi.org/learning/library/complexity-management-projects-programmes-portfolios-11141))

**Industry Analysis:** - Basecamp’s success demonstrates that simplicity in project management tools resonates with teams who don’t want to figure out where to find information ([Medium - Basecamp Power and Pitfalls](https://funficient.medium.com/unpacking-the-power-and-pitfalls-of-project-management-with-basecamp-63a783364fa2)) - Performance issues reported with traditional tools during heavy usage and limitations in complex project workflows ([SmartTask Basecamp Review](https://www.smarttask.io/blog/basecamp-project-management), [Digital Project Manager Basecamp Review](https://thedigitalprojectmanager.com/tools/basecamp-review/)) - Expert analysis confirms that many tools lack functionality required for small IT development teams while being overly complex ([Springman Consulting Basecamp Review](https://springmanconsulting.com/detailed-review-of-basecamp-project-management-tool/))

**Market Trends:** - Growing demand for flexible project management tools that work for small businesses ([Startup.Unitel - Best PM Tools for Small Business](https://startup.unitelvoice.com/best-project-management-tools)) - Latest 2025 analysis of 25+ project management tools highlighting integration and complexity challenges ([Digital Project Manager - Best PM Software 2025](https://thedigitalprojectmanager.com/tools/best-project-management-software/))

### The Gap in the Market

The research clearly shows there’s a significant gap between what traditional project management tools offer and what small teams actually need. Most tools fall into two categories:

1. **Over-Engineered Enterprise Tools**: Designed for large organizations with complex workflows, multiple departments, and dedicated training resources.
2. **Overly Simple Tools**: Basic task lists that lack the structure and features needed for effective project management.

What’s missing is a tool that provides the right level of functionality for small teams - sophisticated enough to handle real project management needs but simple enough to start using immediately without training.

This gap is particularly problematic for: - Student teams working on semester projects - Small startups building their first products - Freelancers managing multiple client projects - Small agencies coordinating team work - Open source projects with distributed contributors

### The Solution Approach

Based on this research, it becomes clear that the solution isn’t to add more features or create more complexity. Instead, the approach should focus on:

**Simplicity by Design**: Every feature should have a clear purpose and immediate value.

**Quick Onboarding**: Users should be productive within minutes, not hours or days.

**Essential Features Only**: Include only the features that 90% of small teams actually use 90% of the time.

**Beautiful, Intuitive Interface**: Good design reduces cognitive load and makes complex tasks feel simple.

**Affordable Pricing**: Pricing models that work for students, startups, and small teams.

This research-backed understanding of the problems in the current market directly informed the design and development of Prism as a solution that addresses these real-world pain points.

## Objectives

1. Create a user-friendly project management system that supports multiple organizations
2. Implement sprint planning functionality for time-boxed work cycles
3. Develop a Kanban board for visual task management with drag-and-drop functionality
4. Build a comprehensive issue tracking system with priorities and status workflows
5. Design a modern, responsive UI with animations and theme support
6. Implement secure authentication and role-based access control
7. Solve the complexity problem by focusing on essential features with elegant design
8. Provide an affordable solution suitable for students, startups, and small teams

## Requirements

### Functional Requirements

1. **User Authentication**
   * User registration and login using Clerk authentication
   * Role-based access control within organizations
   * Protected routes and secure API endpoints
2. **Organization Management**
   * Create and manage multiple organizations
   * Admin-only permissions for sensitive operations
   * User role management within organizations
3. **Project Management**
   * Create and manage projects within organizations
   * Project details including name, key, and description
   * Admin-only project creation and deletion
4. **Sprint Planning**
   * Create sprints with name, start date, and end date
   * Manage sprint status (Planned, Active, Completed)
   * Assign issues to sprints
5. **Issue Tracking**
   * Create and manage issues with title, description, status, and priority
   * Assign issues to users
   * Track issue reporters and assignees
   * Support for markdown in issue descriptions
6. **Kanban Board**
   * Drag-and-drop issue management
   * Visual status columns (Todo, In Progress, In Review, Done)
   * Real-time updates and animations

### Non-Functional Requirements

1. **Performance**
   * Fast loading times and responsive UI
   * Efficient database queries using Prisma ORM
2. **Usability**
   * Intuitive and user-friendly interface
   * Responsive design for all devices
   * Dark/Light theme support
   * Zero learning curve for basic functionality
3. **Security**
   * Secure authentication using Clerk
   * Protected API endpoints
   * Role-based access control
4. **Scalability**
   * Support for multiple organizations and projects
   * Efficient database design for growing data
5. **Maintainability**
   * Well-structured codebase
   * Component-based architecture
   * Server actions for backend operations
6. **Affordability**
   * Cost-effective solution for small teams and students
   * Efficient resource usage to keep operational costs low

## Design

### User Flow

1. **Authentication Flow**:
   * User signs up or logs in using Clerk authentication
   * After authentication, user is redirected to the home page
2. **Organization Flow**:
   * User creates or selects an organization
   * Organization admins can manage projects and users
3. **Project Flow**:
   * Admin creates a new project within an organization
   * Users can view and interact with projects based on their roles
4. **Sprint Flow**:
   * Admin creates sprints with start and end dates
   * Sprints progress through statuses: Planned → Active → Completed
5. **Issue Flow**:
   * Users create issues with details, priority, and status
   * Issues can be assigned to users and linked to sprints
   * Issues progress through statuses: Todo → In Progress → In Review → Done

### Database Design

The database schema includes the following main entities:

1. **User**:
   * Stores user information linked to Clerk authentication
   * Tracks created and assigned issues
2. **Project**:
   * Belongs to an organization
   * Contains sprints and issues
   * Has name, key, and description
3. **Sprint**:
   * Belongs to a project
   * Has name, start date, end date, and status
   * Contains issues
4. **Issue**:
   * Belongs to a project and optionally a sprint
   * Has title, description, status, priority, order
   * Linked to reporter and optional assignee

The schema includes appropriate relationships and indexes for efficient queries.

### Screen Design

1. **Home Page**:
   * Landing page with features and information
   * Navigation to authentication and main application
2. **Authentication Screens**:
   * Sign-in and sign-up pages powered by Clerk
3. **Organization Page**:
   * List of projects within the organization
   * Organization management options for admins
4. **Project Page**:
   * Sprint creation form
   * List of sprints with status indicators
   * Navigation to sprint boards
5. **Sprint Board**:
   * Kanban board with status columns
   * Issue cards with priority indicators
   * Drag-and-drop functionality
6. **Issue Details Dialog**:
   * Detailed view of issue information
   * Options to update status, priority, and other details
   * Delete functionality for authorized users

### API Design

The application uses Next.js Server Actions for backend functionality:

1. **Organization Actions**:
   * getOrganization: Retrieve organization details
   * getOrganizationUsers: Get users within an organization
2. **Project Actions**:
   * createProject: Create a new project
   * getProjects: Retrieve projects for an organization
   * getProject: Get details of a specific project
3. **Sprint Actions**:
   * createSprint: Create a new sprint
   * updateSprintStatus: Update sprint status
4. **Issue Actions**:
   * getIssuesForSprint: Retrieve issues for a sprint
   * createIssue: Create a new issue
   * updateIssueOrder: Update issue order for drag-and-drop
   * deleteIssue: Delete an issue
   * updateIssue: Update issue details

All API endpoints include authentication checks and appropriate error handling.

## Tech Stack and Implementation

### Frontend

* **Next.js 15**: React framework with App Router for routing and server components
* **React 19**: UI library for building component-based interfaces
* **Tailwind CSS 4**: Utility-first CSS framework for styling
* **Framer Motion**: Animation library for smooth transitions
* **Aceternity UI + ShadCn**: Accessible component primitives
* **Lucide React**: Icon library

### Backend & Database

* **Prisma**: ORM for database operations
* **NeonDB with PostgreSQL**: Cloud-based PostgreSQL database
* **Next.js Server Actions**: Backend API functionality

### Authentication & Security

* **Clerk**: Authentication and user management
* **Role-based access control**: Organization-based permissions

### UI Components & Libraries

* **@hello-pangea/dnd**: Drag and drop functionality for Kanban board
* **React Hook Form**: Form management
* **Zod**: Schema validation
* **Sonner**: Toast notifications
* **Vaul**: Drawer components
* **Date-fns**: Date manipulation
* **React Day Picker**: Date picker component
* **MDEditor**: Markdown editor for issue descriptions

## Deployment

The application is deployed on Vercel using GitHub integration with built-in CI/CD pipeline. This setup provides:

1. **Continuous Integration**: Automatic testing and building on code changes
2. **Continuous Deployment**: Automatic deployment when changes are pushed to the main branch
3. **Preview Deployments**: Preview environments for pull requests
4. **Global CDN**: Fast content delivery through Vercel’s global CDN
5. **Serverless Functions**: Efficient execution of server-side code

The deployment process is streamlined and requires minimal manual intervention, making it easy to update the application with new features and bug fixes.

## Conclusion

Prism is a comprehensive project management platform that combines modern web technologies with user-friendly design to create an efficient tool for team collaboration. The application’s organization-based architecture, sprint planning capabilities, and intuitive issue tracking make it suitable for various team sizes and project types, while specifically addressing the complexity problems that plague traditional project management tools.

By focusing on the real needs of small teams, startups, and students, Prism provides a solution that is both powerful enough to handle serious project management needs and simple enough to start using immediately. The research-backed approach to identifying and solving existing problems in the market ensures that Prism addresses genuine pain points rather than adding unnecessary complexity.

The use of Next.js 15 with App Router, Prisma ORM, and Clerk authentication provides a solid foundation for the application, while the modern UI components and animations create an engaging user experience. The deployment on Vercel ensures reliable performance and easy updates.

This project demonstrates the effective use of modern web development practices and tools to create a feature-rich application that meets the real needs of project management teams, particularly those who have been underserved by traditional, overly complex solutions.