# Portfolio Command Center 2.0

A modern, fully functional portfolio website with project management capabilities, AI integration, and secure environment configuration.

## 🚀 Features

### Core Functionality

* **Portfolio Showcase**: Professional portfolio display with project galleries
* **Project Management**: Comprehensive task tracking and milestone management
* **AI Integration**: Gemini AI-powered insights and recommendations
* **Journal System**: Progress tracking and reflection with mood analytics
* **Dashboard Analytics**: Visual progress tracking with Chart.js
* **Responsive Design**: Mobile-first approach with Tailwind CSS

### Technical Highlights

* **Modern Architecture**: Modular React components with TypeScript
* **Secure Configuration**: Environment variables for all API keys
* **State Management**: Context API with useReducer for complex state
* **Firebase Integration**: Authentication and Firestore database
* **Progressive Web App**: Optimized for performance and accessibility

## 🛠️ Technology Stack

* **Frontend**: React 18 + TypeScript + Vite
* **Styling**: Tailwind CSS + Custom CSS
* **Charts**: Chart.js + React Chart.js 2
* **Backend**: Firebase (Auth + Firestore)
* **AI**: Google Gemini API
* **Icons**: Lucide React
* **State**: React Context + useReducer

## 📦 Installation

### Prerequisites

* Node.js 18+ and pnpm
* Firebase project with Firestore enabled
* Google Gemini API key (optional for AI features)

### Quick Start

1. **Clone and Install**

* cd /workspace/portfolio-command-center-rebuild  
  pnpm install

1. **Environment Configuration**

* cp .env.example .env
* Edit .env with your actual values:
* # Firebase Configuration  
  VITE\_FIREBASE\_API\_KEY=your\_firebase\_api\_key  
  VITE\_FIREBASE\_AUTH\_DOMAIN=your\_project.firebaseapp.com  
  VITE\_FIREBASE\_PROJECT\_ID=your\_project\_id  
  VITE\_FIREBASE\_STORAGE\_BUCKET=your\_project.appspot.com  
  VITE\_FIREBASE\_MESSAGING\_SENDER\_ID=your\_sender\_id  
  VITE\_FIREBASE\_APP\_ID=your\_app\_id  
    
  # Google Gemini API (Optional)  
  VITE\_GEMINI\_API\_KEY=your\_gemini\_api\_key

1. **Development Server**

* pnpm dev

1. **Production Build**

* pnpm build

## 🔧 Configuration

### Firebase Setup

1. Create a Firebase project at [Firebase Console](https://console.firebase.google.com)
2. Enable Authentication and Firestore Database
3. Copy your project configuration to the .env file
4. Configure Firestore security rules (see firestore.rules)

### Gemini AI Setup (Optional)

1. Get API key from [Google AI Studio](https://makersuite.google.com/app/apikey)
2. Add to .env file as VITE\_GEMINI\_API\_KEY
3. AI features will be disabled gracefully if not configured

## 🌐 Deployment

### Automatic Deployment

This project includes optimized build configuration for modern deployment platforms:

pnpm build # Creates production build in dist/

### Platform-Specific Guides

**Vercel (Recommended)** - Connect GitHub repository - Environment variables auto-detected from .env.example - Zero-config deployment

**Netlify** - Build command: pnpm build - Publish directory: dist - Add environment variables in site settings

**Firebase Hosting**

npm install -g firebase-tools  
firebase init hosting  
firebase deploy

## 📁 Project Structure

src/  
├── components/ # React components  
│ ├── Dashboard.tsx # Analytics dashboard  
│ ├── Navigation.tsx # Main navigation  
│ ├── Portfolio.tsx # Portfolio showcase  
│ ├── Projects.tsx # Project management  
│ ├── Journal.tsx # Progress journal  
│ └── Onboarding.tsx # User onboarding  
├── contexts/ # React contexts  
│ └── AppContext.tsx # Global state management  
├── hooks/ # Custom React hooks  
│ └── useInitialData.ts  
├── services/ # External service integrations  
│ ├── firebase.ts # Firebase configuration  
│ ├── gemini.ts # AI service  
│ └── sampleData.ts # Demo data  
└── App.tsx # Main application component

## 🔐 Security Features

### Environment Variables

* All API keys stored in environment variables
* No hardcoded secrets in source code
* .env.example template for easy setup

### Firebase Security

* Authentication-based access control
* Firestore security rules
* Secure API key management

### Best Practices

* TypeScript for type safety
* Input validation and sanitization
* Error boundaries for graceful failures
* CSP headers and security headers

## 🎨 Design System

### Color Palette

* **Primary**: Blue to Violet gradient (from-blue-500 to-violet-500)
* **Background**: Dark slate gradient (from-slate-900 via-slate-800 to-slate-900)
* **Text**: Slate color variants for hierarchy
* **Accents**: Emerald, amber, red for status indicators

### Typography

* **Font**: Inter (Google Fonts)
* **Weights**: 400, 500, 600, 700
* **Responsive**: Clamp-based fluid typography

### Components

* Glass morphism effects with backdrop blur
* Smooth transitions and micro-interactions
* Consistent spacing using Tailwind CSS scale
* Accessible color contrasts (WCAG compliant)

## 🧪 Development

### Available Scripts

* pnpm dev - Development server with HMR
* pnpm build - Production build
* pnpm preview - Preview production build
* pnpm lint - ESLint code checking

### Code Style

* TypeScript strict mode enabled
* ESLint with React and TypeScript rules
* Prettier formatting (configured via ESLint)
* Path aliases (@/ for src/)

## 📊 Features Overview

### Dashboard

* Project completion analytics
* Progress visualization with Chart.js
* AI-generated status updates
* Recent activity summary

### Projects

* Full CRUD operations for projects
* Milestone and subtask management
* Progress tracking and analytics
* AI-powered project ideas generation

### Portfolio

* Professional project showcase
* Technology filtering and categorization
* Responsive image galleries
* Contact and resume download

### Journal

* Mood tracking and analytics
* Tag-based organization
* AI insights generation
* Search and filtering capabilities

### Onboarding

* Guided user setup process
* Personalized AI recommendations
* Progressive disclosure of features
* Profile customization

## 🤝 Contributing

1. Fork the repository
2. Create a feature branch (git checkout -b feature/amazing-feature)
3. Commit changes (git commit -m 'Add amazing feature')
4. Push to branch (git push origin feature/amazing-feature)
5. Open a Pull Request

## 📄 License

This project is licensed under the MIT License - see the <LICENSE> file for details.

## 🙏 Acknowledgments

* React team for the amazing framework
* Tailwind CSS for the utility-first approach
* Firebase for backend services
* Google for Gemini AI capabilities
* Chart.js for beautiful data visualization

## 📞 Support

For support, questions, or feature requests: - Open an issue on GitHub - Contact: [your-email@example.com] - Documentation: [project-docs-url]

**Portfolio Command Center 2.0** - Built with ❤️ using modern web technologies