🎨 AI SDK UI Library

A comprehensive React component library for building AI-powered applications. Built with Next.js, TypeScript, and Tailwind CSS, featuring production-ready components optimized for AI/ML interfaces and data-rich applications.

✨ Technology Stack

This library is built with:

🎯 Core Framework

- ⚡ **Next.js 15** - The React framework for production with App Router
- 📘 **TypeScript 5** - Type-safe JavaScript for better developer experience
- 🎨 **Tailwind CSS 4** - Utility-first CSS framework for rapid UI development

🧩 UI Components & Styling

- 🧩 **shadcn/ui** - High-quality, accessible components built on Radix UI
- 🎯 **Lucide React** - Beautiful & consistent icon library
- 🌈 **Framer Motion** - Production-ready motion library for React
- 🎨 **Next Themes** - Perfect dark mode in 2 lines of code

📋 Forms & Validation

- 🪝 **React Hook Form** - Performant forms with easy validation
- ✅ **Zod** - TypeScript-first schema validation

🔄 State Management & Data Fetching

- 🐻 **Zustand** - Simple, scalable state management
- 🔄 **TanStack Query** - Powerful data synchronization for React
- 🌐 **Axios** - Promise-based HTTP client

🗄 Database & Backend

- 🗄 **Prisma** - Next-generation Node.js and TypeScript ORM
- 🔐 **NextAuth.js** - Complete open-source authentication solution

🎨 Advanced UI Features

- 📊 **TanStack Table** - Headless UI for building tables and datagrids
- 🪟 **DND Kit** - Modern drag and drop toolkit for React
- 📊 **Recharts** - Redefined chart library built with React and D3
- 🖼 **Sharp** - High performance image processing

🌍 Internationalization & Utilities

- 🌍 **Next Intl** - Internationalization library for Next.js
- 📅 **Date-fns** - Modern JavaScript date utility library
- 🪝 **ReactUse** - Collection of essential React hooks for modern development

🎯 Why This Library?

- 🤖 **AI-Optimized** - Components specifically designed for AI/ML application interfaces
- 🎨 **Beautiful UI** - Built on shadcn/ui with advanced interactions and animations
- 🔒 **Type Safety** - Fully typed with TypeScript and Zod validation
- 📱 **Responsive** - Mobile-first design principles with smooth animations
- 📊 **Data-Rich** - Specialized components for charts, tables, and data visualization
- 📚 **Well Documented** - Comprehensive Storybook documentation with live examples
- 🌍 **i18n Ready** - Multi-language support built-in
- ⚡ **Performance** - Optimized for production with tree-shaking and code splitting
- 🎨 **Themeable** - Dark mode and customizable theming out of the box
- ♿ **Accessible** - WCAG compliant components built on Radix UI

🚀 Quick Start

Development

```
# Install dependencies
yarn install

# Start Next.js development server
yarn dev

# Start Storybook component explorer
yarn storybook
```

- **Next.js App**: http://localhost:3000 - Demo application
- **Storybook**: http://localhost:6006 - Component documentation

Building

```
# Build Next.js app
yarn build:next

# Build Storybook
yarn build:storybook

# Build both
yarn build
```

🛳️ Deployment to Vercel

This project is configured for single-project deployment to Vercel with path-based routing:

- **Main app**: Deployed at your root domain (e.g., `yourapp.com`)
- **Storybook**: Automatically available at `/storybook` path (e.g., `yourapp.com/storybook`)

Deploy Steps:

1. **Push to GitHub** - Commit and push your changes

   ```
   git add .
   git commit -m "Ready for deployment"
   git push
   ```

2. **Connect to Vercel**

   - Visit vercel.com and import your repository
   - Vercel will auto-detect the Next.js framework

3. **Configure Environment Variables** (if needed)

   - Add any required environment variables in Vercel dashboard
   - Include database connection strings, API keys, etc.

4. **Deploy**

   - Click "Deploy" - Vercel will run `yarn build` which builds both Next.js and Storybook
   - Your app will be live at `your-project.vercel.app`
   - Storybook will be accessible at `your-project.vercel.app/storybook`

Using Vercel CLI:

```
# Install Vercel CLI globally
npm i -g vercel

# Deploy to preview
vercel

# Deploy to production
vercel --prod
```

Build Configuration:

The build process automatically:

1. Builds the Next.js application to `.next/`
2. Builds Storybook to `public/storybook/`
3. Next.js serves Storybook static files from the public directory

Both the main app and Storybook are deployed in a single Vercel project, simplifying deployment and management.