🎨 Design System Assignment

Focus Areas: Color System · UI Components
Tech Stack: React · TypeScript · TailwindCSS (or styled-components) · Storybook

🚀 What You'll Do

Build a small, scalable design system that showcases your ability to create **enterprise-grade, reusable UI components** using React and TypeScript. You'll document them using Storybook and structure the project with scalability in mind.

🎯 Objectives

We're evaluating your ability to:

- Design and implement a robust **token-based color system**
- Build **accessible**, **responsive**, and **reusable** UI components
- Apply **interaction states** and **behavioral logic**
- Document clearly with **Storybook** for developer usability

🧱 Foundational Requirement: Color System

Create a comprehensive token-based color system tailored for B2B/enterprise UI needs:

Include:

- Primary, Secondary, Tertiary colors
- Neutral palette (greys, white, borders, backgrounds)
- Semantic colors: Success, Info, Warning, Error
- Surface/Background layering
- Support for **light & dark theme**
- WCAG-compliant contrast ratios

Requirements:

- Use **CSS variables** or **theme utilities**

- Document usage in **Storybook** with:
    - Token naming conventions
    - Example applications
    - Accessibility considerations

📦 Choose 2 of 5 Components to Build (Fully Functional)

Build **minimum two** of the following UI components. Ensure it is responsive, accessible, and adheres to best practices in state management, interaction handling, and reusability.

📊 Option 1: Advanced Data Table

- Sortable columns with arrow indicators
- Hoverable rows with subtle elevation
- Loading skeleton states
- Pagination controls
- Checkbox selection (including indeterminate state)
- Expandable rows (with smooth animations)
- Filter dropdowns in column headers
- Mobile responsiveness (stacked layout on small screens)

🧭 Option 2: Multi-Step Wizard Form

- Progress indicator with step numbers/icons
- Step titles with checkmarks for completed steps
- Animated transitions between steps
- Navigation (Next, Previous, Skip)
- Validation and error handling
- Async loading states
- Responsive layout
- Optional steps support

📁 Option 3: Advanced File Upload

- Drag & drop zone with feedback
- File preview thumbnails (images/docs)
- Upload progress bars per file
- Error states for failed validations
- Retry, remove all, and bulk actions
- Optional image compression preview

- Mobile camera input support

🗒️ Option 4: Dynamic Form Builder

- Grid-based responsive layout
- Real-time field validation feedback
- Conditional rendering of fields
- Grouping of fields with section headers
- Progress indicator based on form completion
- Auto-save indicator during changes
- Error summary panel at top of form
- Mobile-optimized input types (e.g., date, tel, number)

🪟 Option 5: Advanced Modal System

- Multiple animation types (fade, scale, slide-in)
- Optional backdrop blur effect
- Configurable modal sizes (small, medium, fullscreen)
- Support for stacked modals
- Drawer-style variant (slide from left/right)
- Loading states with built-in spinner overlay
- Keyboard navigation (ESC to close, tab-trapping)

📘 Storybook Documentation Checklist

For **each component**, include a detailed Storybook entry with:

✅ Component name & description
✅ Props & API definitions (with TypeScript types)
✅ Use cases & real-world examples
✅ Anatomy/structure breakdown
✅ States & variants
✅ Interaction behavior
✅ Accessibility notes (ARIA roles, keyboard nav, focus)
✅ Theming and responsiveness handling
✅ Best practices, do’s & don’ts

📁 Project Structure

- Each component in its **own module/folder**
- Use **TypeScript interfaces** for props
- Follow scalable, reusable component patterns
- Apply best practices in file structure, naming, and styling

📤 Submission Guidelines

Please submit the following:

1. **GitHub Repository**
    - Clear folder structure
    - README with setup instructions
    - Description of your approach

2. **Storybook Preview Link**
    - Use Chromatic or Vercel to deploy

3. **GIFs or Screenshots** (optional but recommended)
    - Showcase interactive states or animations