🎨 React Component Development Assignment

Focus Area: UI Components
Tech Stack: React · TypeScript · TailwindCSS · Storybook

🚀 Task

Build **two React components** in 2 days using React, TypeScript and modern patterns. You'll document them using Storybook and structure the project with scalability in mind.

🎯 Component 1: InputField

Create a flexible input component with validation states.

Features:

- Text input with label, placeholder, helper text, error message
- States: disabled, invalid, loading
- Variants: filled, outlined, ghost
- Sizes: small, medium, large
- Optional: clear button, password toggle
- Optional: Support for **light & dark theme**

Props:

```
interface InputFieldProps {
        value?: string;
        onChange?: (e: React.ChangeEvent<HTMLInputElement>) => void;
        label?: string;
        placeholder?: string;
        helperText?: string;
        errorMessage?: string;
        disabled?: boolean;
        invalid?: boolean;
        variant?: 'filled' | 'outlined' | 'ghost';
        size?: 'sm' | 'md' | 'lg';
}
```

📊 Component 2: DataTable

Create a data table with basic functionality.

Features:

- Display tabular data
- Column sorting
- Row selection (single/multiple)
- Loading state
- Empty state

Props:

```
interface DataTableProps<T> {
        data: T[ ];
        columns: Column<T>[ ];
        loading?: boolean;
        selectable?: boolean;
        onRowSelect?: (selectedRows: T[ ]) => void;
}

interface Column<T> {
        key: string;
        title: string;
        dataIndex: keyof T;
        sortable?: boolean;
}
```

✅ Requirements

- TypeScript with proper typing
- Responsive design
- Basic accessibility (ARIA labels)
- Clean, modern styling
- Include basic test

📘 Deliverables

- Two working components

- Simple demo/example usage
- Basic documentation

📤 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