Storybook React Project Documentation

1. Project Overview

This project is a React + TypeScript application scaffolded with Vite and integrated with Storybook for UI component development, testing, and documentation.

2. Folder Structure

```
Storybook-React/
├── .gitignore
├── eslint.config.js
├── index.html
├── package-lock.json
├── package.json
├── README.md
├── tsconfig.app.json
├── tsconfig.json
├── tsconfig.node.json
├── vite.config.ts
├── vitest.config.ts
├── vitest.shims.d.ts
│
├── .storybook/
│   ├── main.ts
│   ├── preview.ts
│   └── vitest.setup.ts
│
├── public/
│   └── vite.svg
│
├── src/
│   ├── App.tsx
│   ├── index.css
│   ├── main.tsx
│   │
│   ├── assets/
│   │   └── react.svg
│   │
│   ├── Components/
│   │   ├── DataTable/
│   │   │   ├── DataTable.tsx
│   │   │   └── DataTable.test.tsx
│   │   │
│   │   └── InputField/
│   │      ├── InputField.tsx
│   │      └── InputField.test.tsx
│   │
│   ├── SampleData/
│   │   └── Data.ts
│   │
│   └── stories/
│      ├── assets/
│      ├── Button.tsx
│      ├── Button.stories.ts
│      ├── Configure.mdx
│      ├── DataTable.stories.tsx
│      ├── Header.tsx
│      ├── Header.stories.ts
│      ├── InputField.stories.tsx
│      ├── Page.tsx
│      ├── Page.stories.ts
│      └── *.css
│
└── storybook-static/
   ├── assets/
   ├── sb-addons/
   ├── sb-common-assets/
   ├── sb-manager/
   ├── iframe.html
   ├── index.html
   └── project.json
```

3. Features

- Vite for fast builds and HMR (Hot Module Replacement)
- Storybook for isolated component development and visual testing
- Vitest for unit and integration testing
- Reusable UI Components (DataTable, InputField etc.)
- Organized folder structure for scalability and maintainability

4. Installation & Setup

1. Clone repo → git clone <repo-url> && cd Storybook-React
2. Install dependencies → npm install
3. Run dev server → npm run dev (http://localhost:5173)
4. Run Storybook → npm run storybook (http://localhost:6006)
5. Build project → npm run build
6. Preview build → npm run preview
7. Run tests → npm run test

5. Scripts

npm run dev → Start Vite dev server
npm run build → Build production-ready app
npm run preview → Preview production build
npm run storybook → Start Storybook in dev mode
npm run build-storybook → Build Storybook static site
npm run test → Run unit tests with Vitest

6. Architecture & Approach

The project follows a component-driven development methodology, leveraging Storybook to design, document, and test components in isolation before integrating them into the main application.

7. Development Workflow

1. Design & document in Storybook
2. Implement component logic
3. Write unit tests colocated with components
4. Integrate into App.tsx after validation
5. Build React app & Storybook separately

8. Testing Strategy

- Unit Tests: Validate isolated component behavior
- Integration Tests: Ensure interoperability
- Visual Testing: Storybook for UI regression checks
- Mocking: SampleData/Data.ts for isolated testing

9. Scalability Considerations

- Reusable design system evolution
- Storybook as living documentation
- Global theming with preview.ts
- CI/CD-ready for both Storybook & App

10. Benefits

- Collaboration between devs & designers
- Maintainability via encapsulated structure
- Confidence with layered testing
- High performance using Vite
- Independent deployment for App & Storybook