

SDP Project Naming Conventions

File names

- Use PascalCase for component file names
- Match the name of the default export
- Examples: `Button.js`, `UserProfile.js`, `NavigationBar.js`

Utilities and Helpers

- Use camelCase for utility file names
- Examples: `formatDate.js`, `apiHelpers.js`, `validationUtils.js`

Styles

- Use the same name as the component, with a `.css`, `.scss`, or `.module.css` extension
- Examples: `Button.css`, `UserProfile.module.css`

Context

- Append `Context` to the name
- Examples: `UserContext.js`, `ThemeContext.js`

Function Names

- Use PascalCase for component functions (they are in fact components)
- Examples: `function Button()`, `function UserProfile()`, `const NavigationBar = () => {}`

Hooks

- Prefix with `use` and use camelCase
- Examples: `useFormValidation()`, `useState()`, `useApiData()`

Event Handlers

- Prefix with `handle` and use camelCase
- Examples: `handleSubmit()`, `handleInputChange()`, `handleUserLogin()`

Utility Functions

- Use camelCase
- Descriptive and action-oriented
- Examples: `formatCurrency()`, `calculateTotalPrice()`, `validateEmail()`

Test Files

Naming

- Use the same name as the file being tested, appended with ``.test``
- Examples: `Button.test.js`.

Test Suite Naming

- Use `describe` to group tests, with a descriptive string
- Example: `describe('Button Component', () => { ... })`

Individual Test Naming

- Use `it` or `test` with a descriptive string in natural language
- Start with 'should' for clarity
- Examples:
 - `it('should render with default props', () => { ... })`
 - `test('should call onClick when clicked', () => { ... })`

Test Function Naming

- For any helper functions in tests, use camelCase
- Examples: `setupComponent()`, `mockApiCall()`, `generateTestUser()`

