**Assignment 3**

**useState Basics — React + Next.js + TypeScript**

Contents

[Overview 2](#_Toc209043581)

[Objectives 2](#_Toc209043582)

[Deliverables 2](#_Toc209043583)

[Functional Requirements 2](#_Toc209043584)

[1) Counter 2](#_Toc209043585)

[2) Text Mirror 2](#_Toc209043586)

[3) Emoji Toggle 3](#_Toc209043587)

[Non‑Functional & Technical Constraints 3](#_Toc209043588)

[Framework & Language 3](#_Toc209043589)

[Component Types 3](#_Toc209043590)

[Project Structure 3](#_Toc209043591)

[Accessibility 3](#_Toc209043592)

[Styling 3](#_Toc209043593)

[TypeScript 4](#_Toc209043594)

[No Networking 4](#_Toc209043595)

[Starter & Run Instructions (to include in your README) 4](#_Toc209043596)

[Acceptance Criteria (Checklist) 4](#_Toc209043597)

[Project Setup 4](#_Toc209043598)

[Counter 4](#_Toc209043599)

[Text Mirror 4](#_Toc209043600)

[Emoji Toggle 5](#_Toc209043601)

[Constraints 5](#_Toc209043602)

[Quality 5](#_Toc209043603)

[Expected Output 5](#_Toc209043604)

[Submission 5](#_Toc209043605)

# Overview

Build a small, single‑page Next.js application (App Router) in TypeScript that demonstrates an interactive UI using only the React useState hook. The goal is to practice client components, state management basics, and clean layout—without introducing other React hooks or external state libraries.

# Objectives

Create a minimal Next.js (TypeScript) app with a simple layout (header, main, footer).  
Implement three independent interactive components using only useState:  
 1) Counter — increment/decrement/reset  
 2) Text Mirror — live preview of text input + greeting  
 3) Emoji Toggle — switch between two emojis and button labels  
Ensure type safety, basic accessibility, and clean structure/styles.

# Deliverables

A Next.js project containing:  
 • app/layout.tsx (global layout with header & footer)  
 • app/page.tsx (renders the three components)  
 • app/components/Counter.tsx  
 • app/components/TextMirror.tsx  
 • app/components/EmojiToggle.tsx  
 • app/globals.css (minimal styling)  
 • package.json, tsconfig.json, next-env.d.ts, .gitignore  
A README with project name, setup & run instructions, and Node.js version used.  
Screenshots showing: counter clicks, typing preview/greeting, and emoji toggle.

# Functional Requirements

## 1) Counter

• Renders the current count (start at 0).  
• Three buttons: -1 (decrement), Reset (sets to 0), +1 (increment).  
• All interactions handled with useState.

## 2) Text Mirror

• A labeled text input (“Type something:”).  
• A Preview line that mirrors the input as you type (show “(nothing yet)” when empty).  
• A greeting line: Hello, <name || "stranger">!

## 3) Emoji Toggle

• Shows one emoji at a time: 🌑 (off) or 🏡 (on).  
• A button toggles states; label changes to “Show house” (off) or “Hide house” (on).

# Non‑Functional & Technical Constraints

## Framework & Language

• Next.js (App Router) with TypeScript.  
• Only the React useState hook for stateful behavior.  
• Do NOT use useEffect, useReducer, useMemo, Context API, or external state managers.

## Component Types

• Components that use state must be Client Components (add "use client" at the top).  
• app/layout.tsx remains a Server Component.

## Project Structure

app/  
 layout.tsx  
 page.tsx  
 components/  
 Counter.tsx  
 TextMirror.tsx  
 EmojiToggle.tsx  
 globals.css

## Accessibility

• Buttons have clear text and/or aria-labels.  
• Emoji region uses aria-live="polite" (announce changes).  
• Inputs are associated with <label> via htmlFor/id.

## Styling

• Use a single global stylesheet: app/globals.css.  
• Keep styling minimal (readable contrast, spacing, simple layout).

## TypeScript

• No any in component props or state—use correct types.  
• Enable/keep TypeScript defaults from Next.js.

## No Networking

• No API calls or data fetching—keep it local and static.

# Starter & Run Instructions (to include in your README)

1) Create project (recommended)

npx create-next-app@latest usestate-basics --typescript

Choose App Router (Yes). ESLint optional. Tailwind not required.

2) Run locally

npm install  
npm run dev  
# Open http://localhost:3000

# Acceptance Criteria (Checklist)

## Project Setup

[ ] Next.js App Router with TypeScript compiles and runs (npm run dev).  
[ ] Directory structure matches the spec.  
[ ] app/layout.tsx imports ./globals.css.

## Counter

[ ] Displays the current count (initially 0).  
[ ] +1 increments; -1 decrements; Reset sets to 0.

## Text Mirror

[ ] Input updates a preview line in real time.  
[ ] Empty input shows (nothing yet) and greets “stranger”.

## Emoji Toggle

[ ] Toggles between 🌑 and 🏡 on each click.  
[ ] Button label changes accordingly (“Show house”/“Hide house”).

## Constraints

[ ] Only useState used for state; no other hooks or state libraries.  
[ ] Client components correctly marked with "use client".

## Quality

[ ] Basic accessibility: labels, aria attributes, readable contrast.  
[ ] TypeScript types present and correct (no any).  
[ ] Code is organized, readable, and lint-clean (or documented).

# Expected Output

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

# Submission

Share a short README with Node version, setup & run steps, and any notes on decisions. Include screenshots demonstrating the three interactions.

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++