# Assignment 2: State and Event Handling

Awareness, Application, Mastery, Influence

Gaurab Wagle

June 9, 2025

## Overview of the Assignment

This assignment focuses on the core concepts of state and event handling in React. The primary goal was to understand and apply React's state and using of custom hooks. Throughout these assignments, I learned a lot about state usage and management.

## GitHub Repository Link

Get the complete code and project details on my GitHub repository: I will soon upload the link of the deployed site.
`https://github.com/gxaurab/Tangible/tree/main/Asg1PoW/src/components/Part2`

## Awareness

- **Rubric:** Use React's `useState` hook to manage simple state, and handle basic events like button clicks.

- **Proof of Work:** Build a toggle button that switches its label between "ON" and "OFF" when clicked, updating state accordingly.

### What I Understood

I remembered the W3Schools JavaScript tutorial about switching the light bulb ON and OFF. I learned it during my +2 days. I tried to replicate the same functionality using React's `useState` hook. The task required me to manage a simple state change—toggling between two values, "ON" and "OFF"—based on user interaction.

## Application

- **Rubric:** Manage multiple state variables and update UI based on user input or events.

- **Proof of Work:** Create a search box that filters a hardcoded list of items as the user types.

### What I Understood

This part of the assignment required me to manage multiple states at once. At first, it took me quie a long time to figure out but later I broke down and understood what it wants me to do.

So,I used a hardcoded list of US cities and implemented a search feature that dynamically filtered the list based on the user's input. The user could type into the search box, and the UI would update in real-time to show only the relevant cities.

### Learning Outcome

Through this exercise, I learned how to manage dynamic states, and how events—such as typing into an input field—can affect the state of multiple variables in React.

## Mastery

- **Rubric:** Coordinate several pieces of state together and handle complex input flows like opening/closing modals.

- **Proof of Work:** Build a modal component that can be opened and closed, with form inputs inside whose values are managed with state.

### What I Understood

At first, i had to understand what a modal was. I watched some youtube videos and read some blogs and then realized about the modal componenet. The modal could be opened with a button click, and the state would change accordingly, showing or hiding the modal.

I handled the state of the input fields within the modal. As the user interacted with these inputs, the state of each field was updated, allowing for dynamic changes to the content inside the modal.

### Learning Outcome

CSS had a key role in making the modal, i also learned about managing state for modal like components. "inset" was the key property in making modal look good.

## Influence

- **Rubric:** Create a `useInput` custom hook that manages input value and `onChange` logic, then use it in multiple inputs.

- **Proof of Work:** Created a custom hook named `useInput` to manage input fields efficiently.

## What I Understood

At this stage, I was introduced to the concept of custom hooks. I built a custom hook called `useInput` to handle the input value and the `onChange` logic for multiple input fields. The custom hook simplified my code and made it reusable across multiple components, which is a big advantage when managing similar state logic in different places.

## Learning Outcome

I spent quite a lot of time reading about he customHooks. I found it very useful and very easy as well because a custom hook means, you are making a function that could use another hooks inside of it and you implement your custom logic into it, make it reusable. It reduces repetition and allow for better code organization.

# Working Example

Below is a screenshot demonstrating the working of all 4 different components.

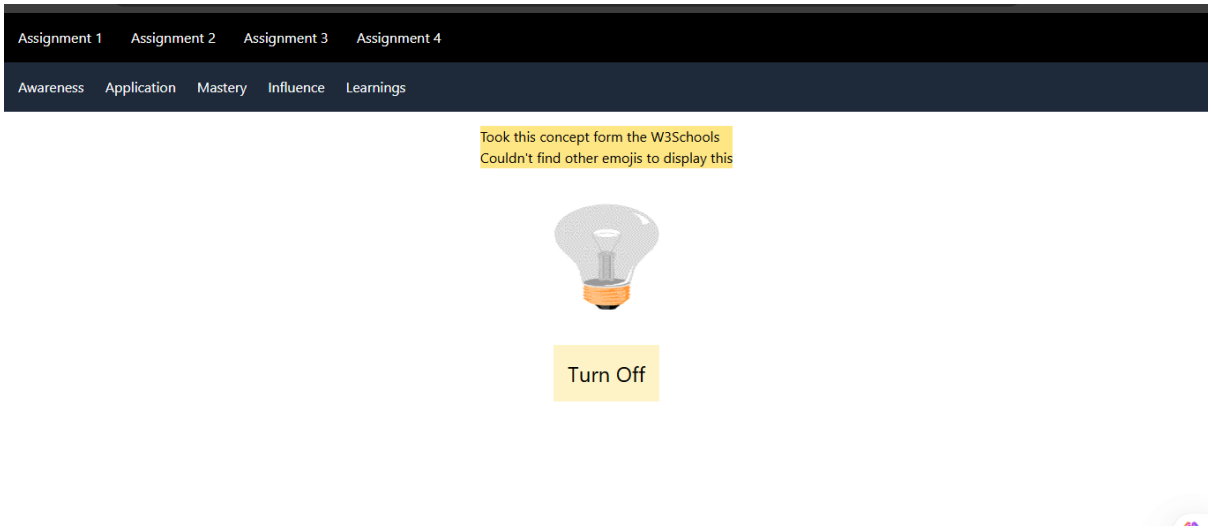


Figure 1: Awareness

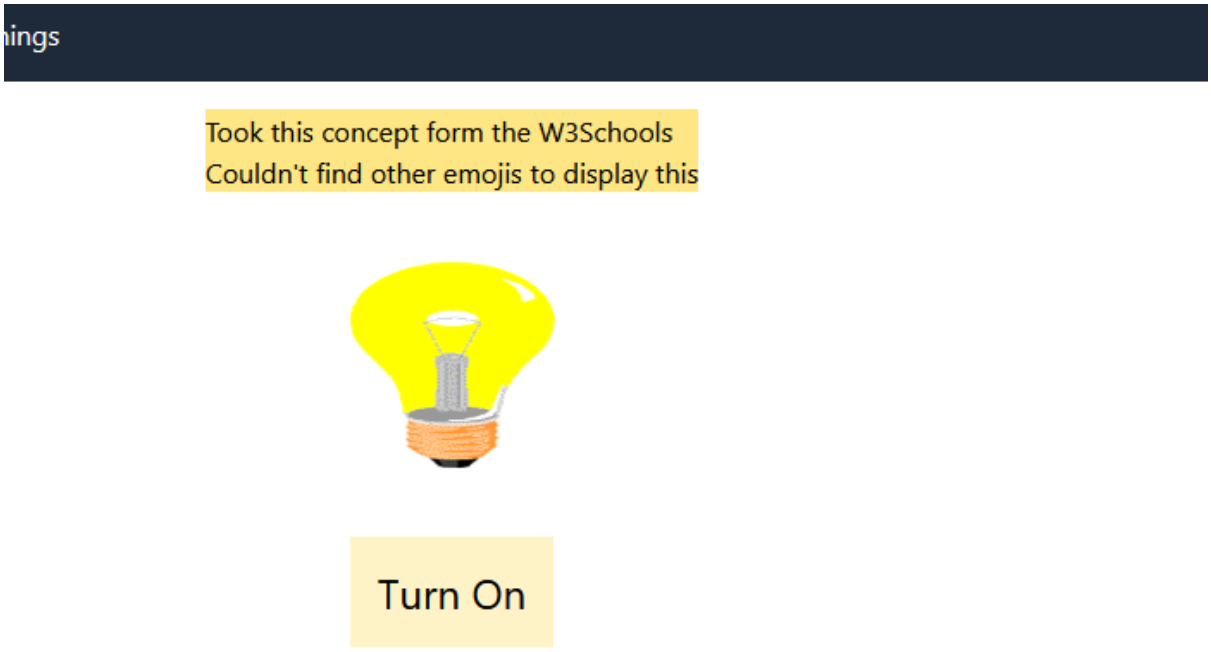


Figure 2: Awareness

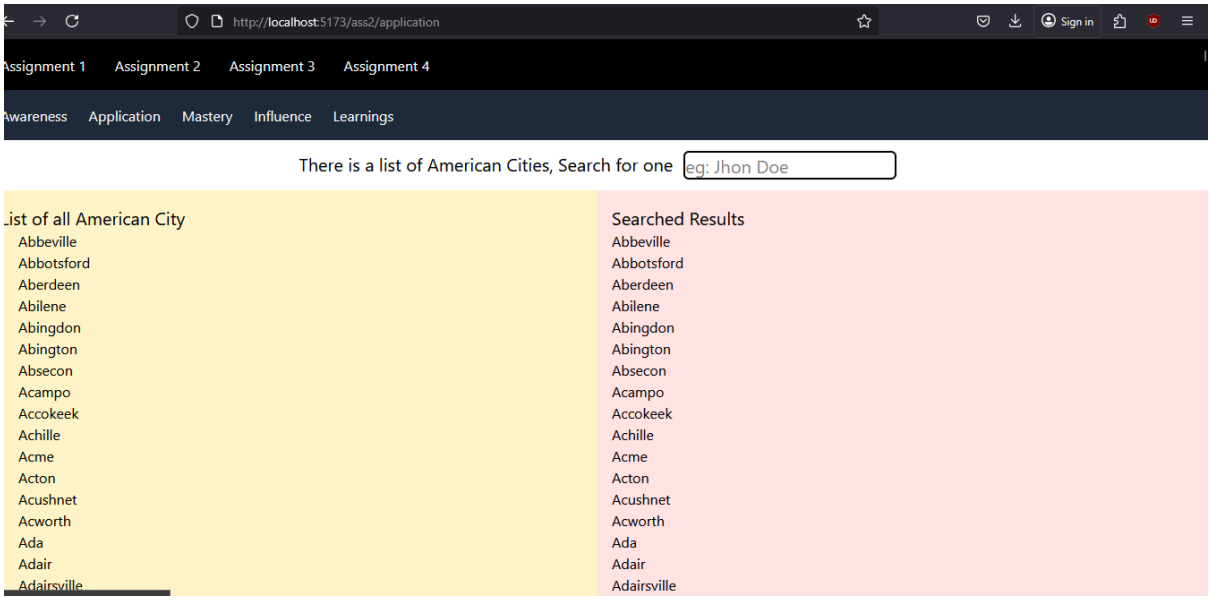


Figure 3: Application

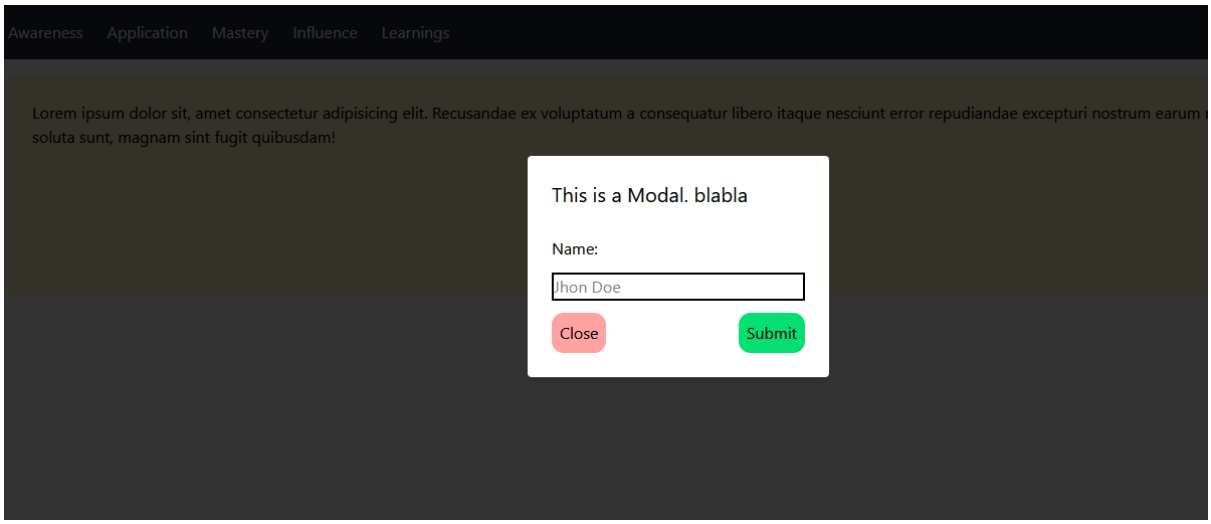


Figure 4: Mastery

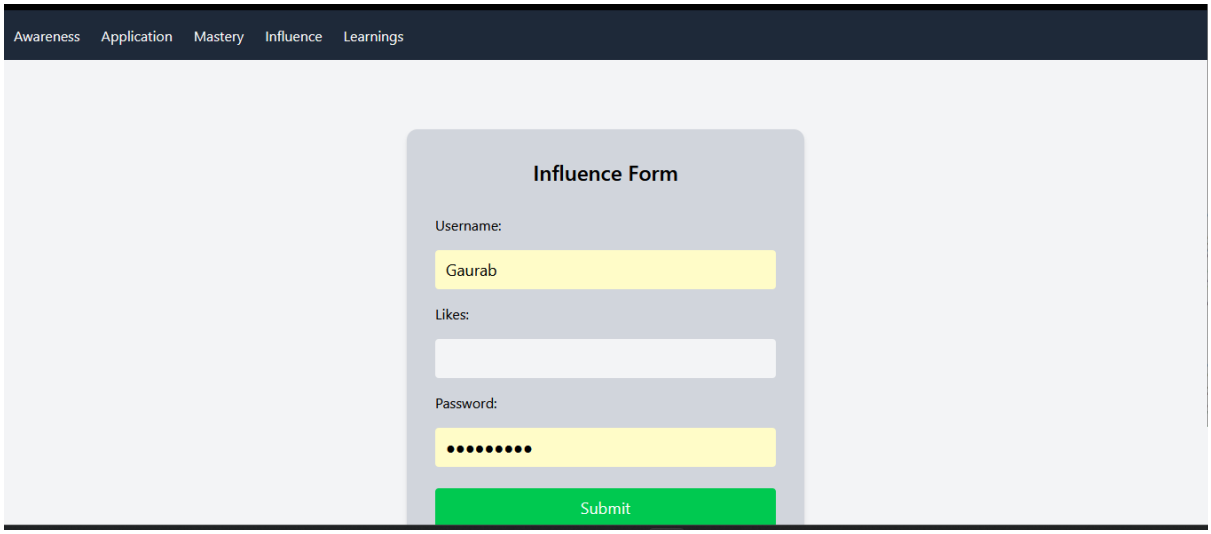


Figure 5: Influence