

# Source A

## Directory Structure

```
.
├── components
│   ├── Modal.tsx
│   ├── Panel.tsx
│   └── Parent.tsx
└── page.tsx
```

## Project Files

===== ./components/Panel.tsx =====

```
'use client'

import React from 'react'

interface PanelProps {
  isExpanded: boolean
  children: React.ReactNode
}

export default function Panel({ isExpanded, children }: PanelProps) {
  return <div style={{ display: isExpanded ? 'block' : 'none' }}>
    {children}</div>
  }
}
```

===== ./components/Modal.tsx =====

```
'use client'

import React, { forwardRef, useState } from 'react'
import Panel from './Panel'

interface ModalProps {
  title: string
  show: boolean
}

const Modal = forwardRef(function Modal({ title, show }: ModalProps, ref: any) {
  const [visible, setVisible] = useState(show)
  const [asyncState, setAsyncState] = useState(false)

  const internalButtonRef = React.createRef<HTMLButtonElement>()
```

```

const openModal = async () => {
  await new Promise((r) => setTimeout(r, 200))

  if (internalButtonRef.current) {
    setAsyncState(true)
    setVisible(true)
    internalButtonRef.current.click()
  }
}

React.useEffect(() => {
  if (show) {
    openModal()
  }
}, [show])

return (
  <Panel isExpanded={visible}>
    <div className="modal">
      <h2>{title}</h2>
      <button ref={internalButtonRef}>Confirm</button>
      {asyncState && <p>Async mode active</p>}
    </div>
  </Panel>
)
})

export default Modal

```

===== ./components/Parent.tsx =====

```

'use client'

import React, { useRef } from 'react'
import Modal from './Modal'

export default function Parent() {
  const modalRef = useRef<{ openModal: () => void }>(null)

  const handleClick = () => {
    modalRef.current?.openModal()
  }

  return (
    <div>
      <h1>Broken Modal Test</h1>
      <button onClick={handleClick}>Open Modal</button>
      <Modal ref={modalRef} title="Async Modal" show={false} />
    </div>
  )
}

```

===== ./page.tsx =====

```
import React from 'react'
import Parent from './components/Parent'

export default function AiPromptPage() {
  return <Parent />
}
```