# Web Accessibility Training Case Study: “MyEdu Portal”

## Background

“MyEdu Portal” is a fictional educational platform designed for students to access courses, submit assignments, and interact with instructors. It was built using React and Angular for different modules.

## Case Study Objectives

* Identify accessibility barriers in the existing platform.
* Apply WCAG 2.2 standards (Levels A & AA).
* Implement fixes in both Angular and React.
* Validate compliance using accessibility tools.

## Accessibility Issues Identified

|  |  |  |
| --- | --- | --- |
| Issue | WCAG Guideline | Description |
| No skip navigation link | 2.4.1 Bypass Blocks | Users using screen readers cannot skip repetitive menus. |
| Inadequate color contrast | 1.4.3 Contrast (Minimum) | Text on buttons and links had low contrast ratios. |
| Missing form labels | 1.3.1 Info and Relationships | Login and registration forms had unlabeled fields. |
| Dynamic content updates without ARIA | 4.1.3 Status Messages | When submitting forms, no screen reader feedback was provided. |
| Focus not managed in modals | 2.4.3 Focus Order | Focus did not move to modal dialogs. |
| Non-descriptive link text | 2.4.4 Link Purpose | Links used vague text like “Click here.” |
| No keyboard navigation | 2.1.1 Keyboard | Navigation menus required mouse interaction only. |

## Fixes and Implementation

### React Module (Course Page)

Problem: Dynamic course load did not inform screen readers.  
Fix:

import { useEffect, useState } from 'react';  
  
function CourseLoad() {  
 const [message, setMessage] = useState('');  
   
 useEffect(() => {  
 setMessage('Course content loaded successfully');  
 }, []);  
  
 return (  
 <>  
 <div role="status" aria-live="polite">  
 {message}  
 </div>  
 </>  
 );  
}

Problem: Poor contrast on course titles.  
Fix:

.course-title {  
 color: #222; /\* Previously #999 \*/  
 background: #fff;  
}

### Angular Module (Login Page)

Problem: No label for username input.  
Fix:

<label for="username">Username</label>  
<input id="username" name="username" type="text" />

Problem: Modal lacks focus trapping.  
Fix:

import { FocusTrap } from '@angular/cdk/a11y';  
  
@ViewChild(FocusTrap) trap: FocusTrap;  
  
openModal() {  
 this.trap.focusInitialElement();  
}

## Testing and Validation Tools

|  |  |
| --- | --- |
| Tool | Purpose |
| axe DevTools | Detects common accessibility issues. |
|  |  |
| WAVE | Web accessibility evaluation tool. |
| Lighthouse | Runs accessibility audits. |
| WCAG Checklist | Manual validation reference. |

## Learning Outcomes

* Understand real-world accessibility problems in web apps.
* Implement WCAG 2.2 standards practically.
* Ensure semantic HTML and ARIA usage.
* Validate accessibility using automated and manual tools.

## Requirements:

1.Developer must have buggy code – shows the WCAG violations

and must have solution code , must compare how they fixed the bug.

2.List out all POUR principles in the code

3.Developer can select any framework (react or angular or vue or plain html with JavaScript) what ever they are comfortable .