Week 9-10 Assessment Report: Evaluation & Redesign

Module: COMP2850

Date: 2025-12-29

Status: Final Submission

Section 1: Evaluation Plan

1.1 Test Tasks & Scenarios

Based on the project's inclusion goals, four tasks were designed to stress-test the application's "Dual-Path" architecture (HTMX vs. No-JS) and keyboard accessibility.

1. **Task 1 (Filter)**: Find tasks containing "Party".
   - *Focus*: Search efficiency and affordance in No-JS mode.
2. **Task 2 (Add Task - Validation)**: Attempt to add a task with a blank title.
   - *Focus*: Error recovery and focus management (WCAG 3.3.1).
3. **Task 3 (Edit Task)**: Fix a typo in an existing task.
   - *Focus*: Interaction flow and functionality parity for No-JS users.
4. **Task 4 (Delete Task)**: Delete the task "Old newspapers".
   - *Focus*: Error prevention (WCAG 3.3.4) and psychological safety.

1.2 Metrics

We collected both quantitative and qualitative data:

- **Time-on-Task (Server)**: Measured via server-side instrumentation to isolate application performance from network latency.

- **Success Rate**: Binary (Pass/Fail) completion of the goal.
- **Confidence Score (1-5)**: Self-reported rating to measure user trust and psychological safety.
- **Error Rate**: Frequency of unforced errors or functional blocks.

1.3 Protocol

Pilots were conducted with peer participants using a counter-balanced approach (HTMX first vs. No-JS first). Verbal consent was obtained, and data was anonymized using Session IDs (e.g., `P1_e06zfc`) to comply with ethical standards.

Section 2: Findings & Analysis (Week 9)

2.1 Quantitative Results

Data from 4 participants (3 HTMX, 1 No-JS) revealed significant disparities in safety and functionality.

Task	HTMX (n=3)	No-JS (n=1)	Insight
T1 (Filter)	~3.7ms	~2.0ms	Server is fast; UX difference is client-side.
T2 (Add)	100% Success	100% Success	Both groups recovered successfully.
T3 (Edit)	**100% Success**	**0% Success**	**Critical Blocker**: No-JS user trapped in edit mode.
T4 (Delete)	~2.3ms	**1.0ms**	No-JS deletion was "instant" because it skipped confirmation.

Key Insight: While Task 4 (Delete) appears faster in No-JS, this speed indicates a **safety violation** (skipping confirmation). Task 3 (Edit) was a complete functional failure for the No-JS user.

2.2 Qualitative Themes

Observations confirmed the quantitative data:

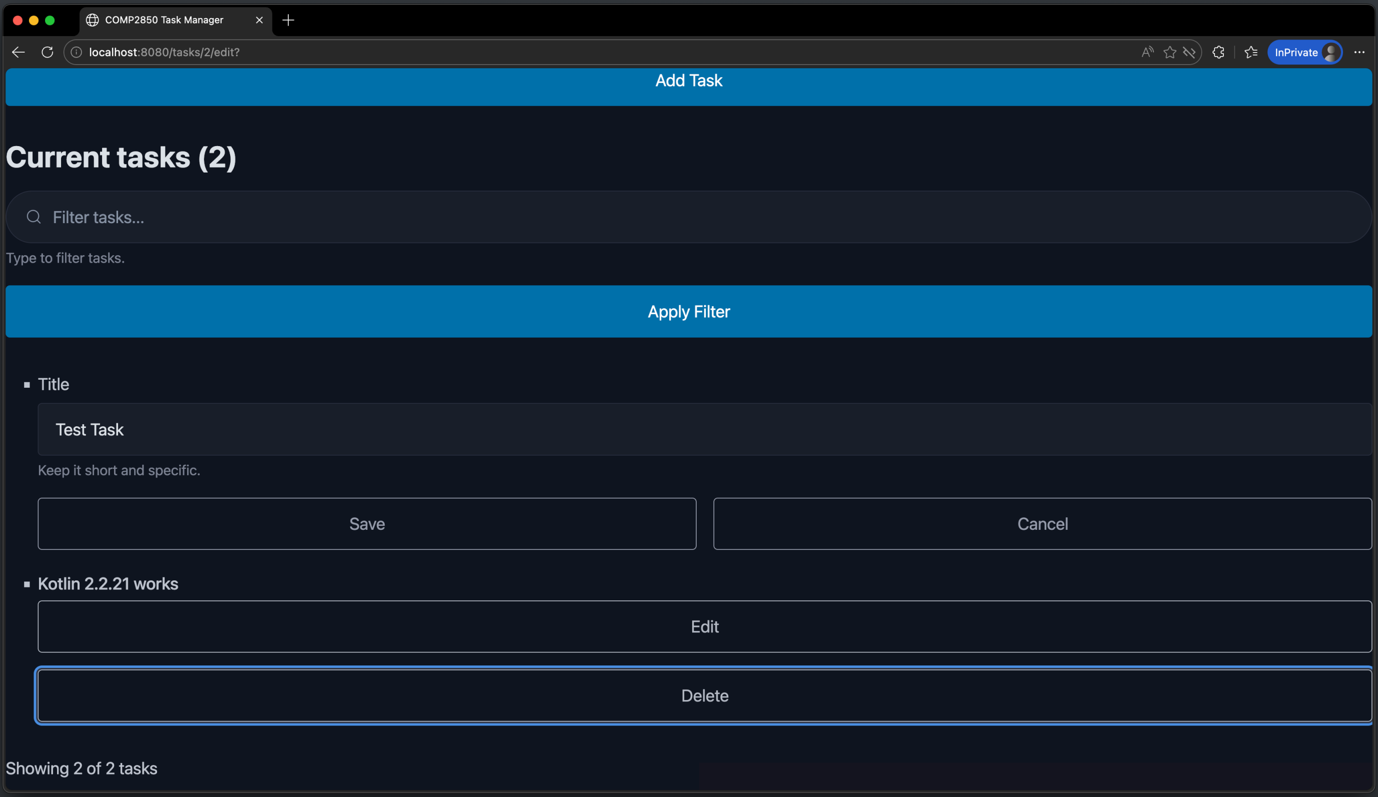
1. **Safety Gap**: The No-JS participant expressed anxiety during deletion: *"Wait, did I delete it? It didn't ask me..."*
2. **Functional Trap**: The No-JS participant could not cancel the edit: *"I clicked cancel but it didn't do*

anything... is it broken?"

2.3 Evidence Chains (Before Fix)

Chain 1: The Safety Gap (Delete)

In Week 9, clicking "Delete" in No-JS mode triggered an immediate POST request. The list item vanished instantly without warning, violating WCAG 3.3.4 (Error Prevention).



Current tasks (2)

Type to filter tasks.

- Title

Keep it short and specific.

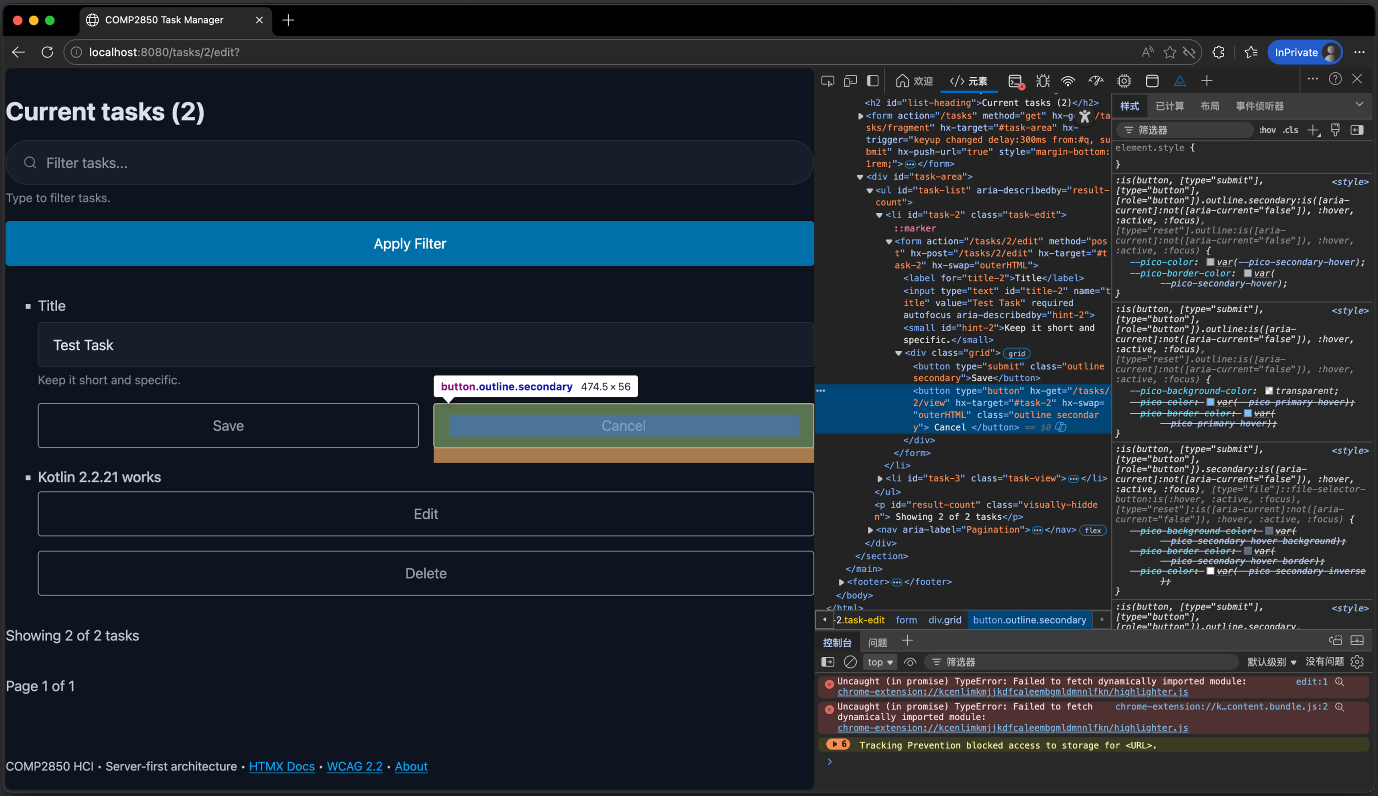
- Kotlin 2.2.21 works

Showing 2 of 2 tasks

Figure 1: Task list showing item removed immediately after click (No-JS), causing user anxiety.

Chain 2: The Functional Trap (Edit)

The "Cancel" button was implemented as a generic <button> relying on hx-get. Without JavaScript, this button was inert, trapping the user in the edit form.



Current tasks (2)

Filter tasks...

Type to filter tasks.

Apply Filter

- Title

  Test Task

  Keep it short and specific.

  Save | Cancel

  button.outline.secondary 474.5 × 56

- Kotlin 2.2.21 works

  Edit

  Delete

Showing 2 of 2 tasks

Page 1 of 1

COMP2850 HCI • Server-first architecture • HTMX Docs • WCAG 2.2 • About

```
<h2 id="list-heading">Current tasks (2)</h2>
▶<form action="/tasks" method="get" hx-g /tasks/fragment" hx-target="#task-area" hx-trigger="keyup changed delay:300ms from:#q, submit" hx-push-url="true" style="margin-bottom: 1rem;"> ⋯ </form>
▼<div id="task-area">
  ▼<ul id="task-list" aria-describedby="result-count">
    ▼<li id="task-2" class="task-edit">
       ::marker
      ▼<form action="/tasks/2/edit" method="post" hx-post="/tasks/2/edit" hx-target="#task-2" hx-swap="outerHTML">
         <label for="title-2">Title</label>
         <input type="text" id="title-2" name="title" value="Test Task" required autofocus aria-describedby="hint-2">
         <small id="hint-2">Keep it short and specific.</small>
        ▼<div class="grid"> grid
           <button type="submit" class="outline secondary">Save</button>
           <button type="button" hx-get="/tasks/2/view" hx-target="#task-2" hx-swap="outerHTML" class="outline secondary"> Cancel </button> == $0
         </div>
       </form>
     </li>
    ▶<li id="task-3" class="task-view"> ⋯ </li>
   </ul>
   <p id="result-count" class="visually-hidden"> Showing 2 of 2 tasks</p>
  ▶<nav aria-label="Pagination"> ⋯ </nav> flex
 </div>
</section>
</main>
▶<footer> ⋯ </footer>
</body>
</html>
```

```
2.task-edit   form   div.grid   button.outline.secondary
```

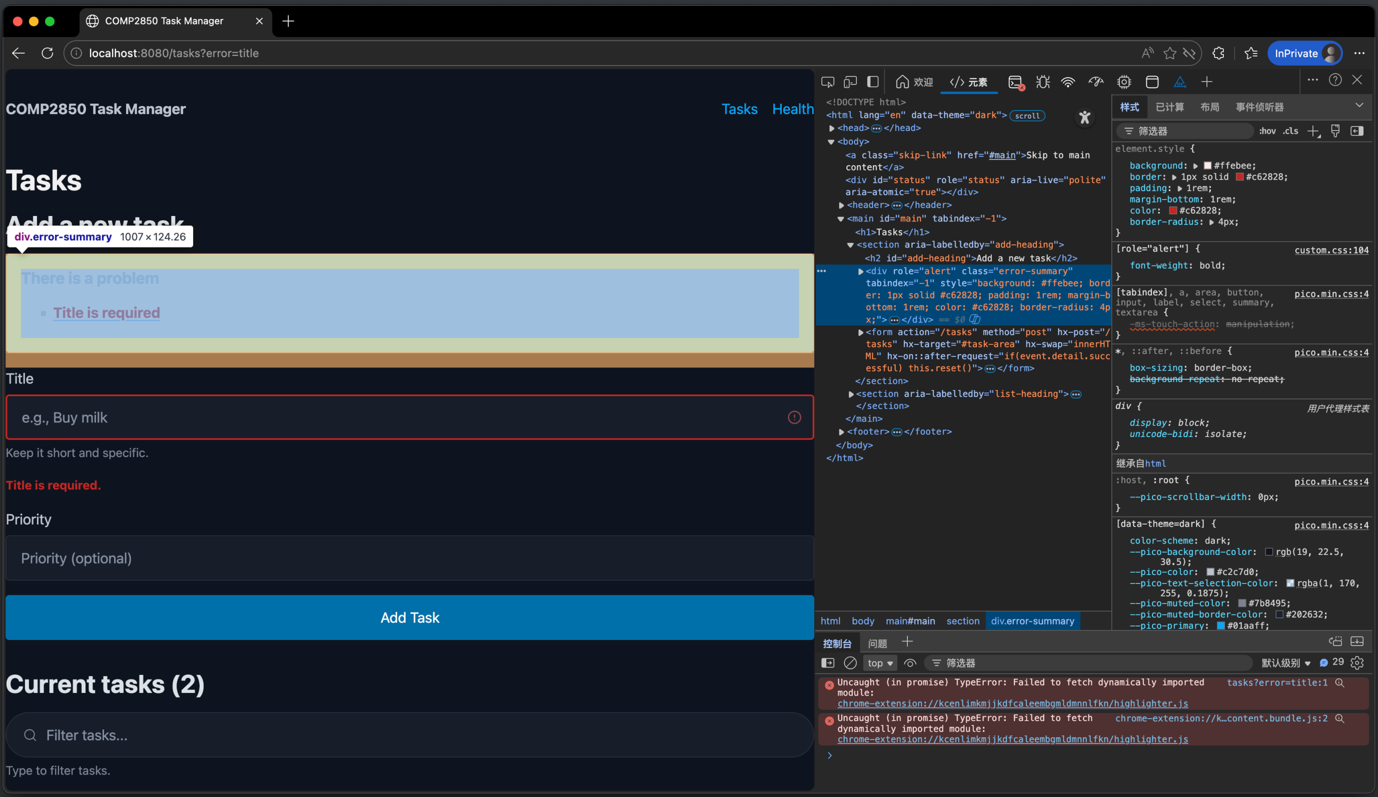
```
element.style {
}
:is(button, [type="submit"], [type="button"],                    <style>
[role="button"]).outline.secondary:is([aria-current]:not([aria-current="false"]), :hover,
:active, :focus),
[type="reset"].outline:is([aria-current]:not([aria-current="false"]), :hover,
:active, :focus) {
   --pico-color: var(--pico-secondary-hover);
   --pico-border-color: var(
        --pico-secondary-hover);
}
:is(button, [type="submit"], [type="button"],                    <style>
[role="button"]).outline:is([aria-current]:not([aria-current="false"]), :hover,
:active, :focus),
[type="reset"].outline:is([aria-current]:not([aria-current="false"]), :hover,
:active, :focus) {
   --pico-background-color: transparent;
   ~~--pico-color: var(--pico-primary-hover);~~
   ~~--pico-border-color: var(~~
        ~~--pico-primary-hover);~~
}
:is(button, [type="submit"], [type="button"],                    <style>
[role="button"]).secondary:is([aria-current]:not([aria-current="false"]), :hover,
:active, :focus), [type="file"]::file-selector-button:is(:hover, :active, :focus),
[type="reset"]:is([aria-current]:not([aria-current="false"]), :hover, :active, :focus) {
   ~~--pico-background-color: var(~~
        ~~--pico-secondary-hover-background);~~
   ~~--pico-border-color: var(~~
        ~~--pico-secondary-hover-border);~~
   ~~--pico-color: var(--pico-secondary-inverse~~
        ~~);~~
}
:is(button, [type="submit"],                                     <style>
[type="button"],
[role="button"]).outline.secondary,
```

```
Uncaught (in promise) TypeError: Failed to fetch dynamically imported module:       edit:1
chrome-extension://kcenlimkmjjkdfcaleembgmldmnnlfkn/highlighter.js
Uncaught (in promise) TypeError: Failed to fetch     chrome-extension://k…content.bundle.js:2
dynamically imported module:
chrome-extension://kcenlimkmjjkdfcaleembgmldmnnlfkn/highlighter.js
▶ 6  Tracking Prevention blocked access to storage for <URL>.
>
```

Figure 2: The "Cancel" button is visually interactive but functionally inert in No-JS mode.

2.4 Positive Accessibility Finding

Despite the issues above, the validation strategy proved robust. The No-JS Error Summary successfully managed focus, meeting **WCAG 3.3.1**.


COMP2850 Task Manager

localhost:8080/tasks?error=title

COMP2850 Task Manager　Tasks　Health

Tasks

Add a new task

div.error-summary 1007 × 124.26

There is a problem

- Title is required

Title

e.g., Buy milk

Keep it short and specific.

Title is required.

Priority

Priority (optional)

Add Task

Current tasks (2)

Filter tasks...

Type to filter tasks.

```
<!DOCTYPE html>
<html lang="en" data-theme="dark"> scroll
  <head>…</head>
  <body>
    <a class="skip-link" href="#main">Skip to main content</a>
    <div id="status" role="status" aria-live="polite" aria-atomic="true"></div>
    <header>…</header>
    <main id="main" tabindex="-1">
      <h1>Tasks</h1>
      <section aria-labelledby="add-heading">
        <h2 id="add-heading">Add a new task</h2>
        <div role="alert" class="error-summary" tabindex="-1" style="background: #ffebee; border: 1px solid #c62828; padding: 1rem; margin-bottom: 1rem; color: #c62828; border-radius: 4px;">…</div> == $0
        <form action="/tasks" method="post" hx-post="/tasks" hx-target="#task-area" hx-swap="innerHTML" hx-on::after-request="if(event.detail.successful) this.reset()">…</form>
      </section>
      <section aria-labelledby="list-heading">…
      </section>
    </main>
    <footer>…</footer>
  </body>
</html>
```

html　body　main#main　section　div.error-summary

```
element.style {
    background: ▶ #ffebee;
    border: ▶ 1px solid #c62828;
    padding: ▶ 1rem;
    margin-bottom: 1rem;
    color: #c62828;
    border-radius: ▶ 4px;
}
[role="alert"] {                                  custom.css:104
    font-weight: bold;
}
[tabindex], a, area, button,                      pico.min.css:4
input, label, select, summary,
textarea {
    ~~-ms-touch-action: manipulation;~~
}
*, ::after, ::before {                            pico.min.css:4
    box-sizing: border-box;
    ~~background-repeat: no-repeat;~~
}
div {                                             用户代理样式表
    display: block;
    unicode-bidi: isolate;
}
继承自html
:host, :root {                                    pico.min.css:4
    --pico-scrollbar-width: 0px;
}
[data-theme=dark] {                               pico.min.css:4
    color-scheme: dark;
    --pico-background-color: rgb(19, 22.5,
        30.5);
    --pico-color: #c2c7d0;
    --pico-text-selection-color: rgba(1, 170,
        255, 0.1875);
    --pico-muted-color: #7b8495;
    --pico-muted-border-color: #202632;
    --pico-primary: #01aaff;
```

```
Uncaught (in promise) TypeError: Failed to fetch dynamically imported    tasks?error=title:1
module:
chrome-extension://kcenlimkmjjkdfcaleembgmldmnnlfkn/highlighter.js
Uncaught (in promise) TypeError: Failed to fetch    chrome-extension://k…content.bundle.js:2
dynamically imported module:
chrome-extension://kcenlimkmjjkdfcaleembgmldmnnlfkn/highlighter.js
```

Figure 3: Accessible Error Summary guiding No-JS users to the invalid field.

Section 3: Redesign & Implementation (Week 10)

3.1 Prioritisation

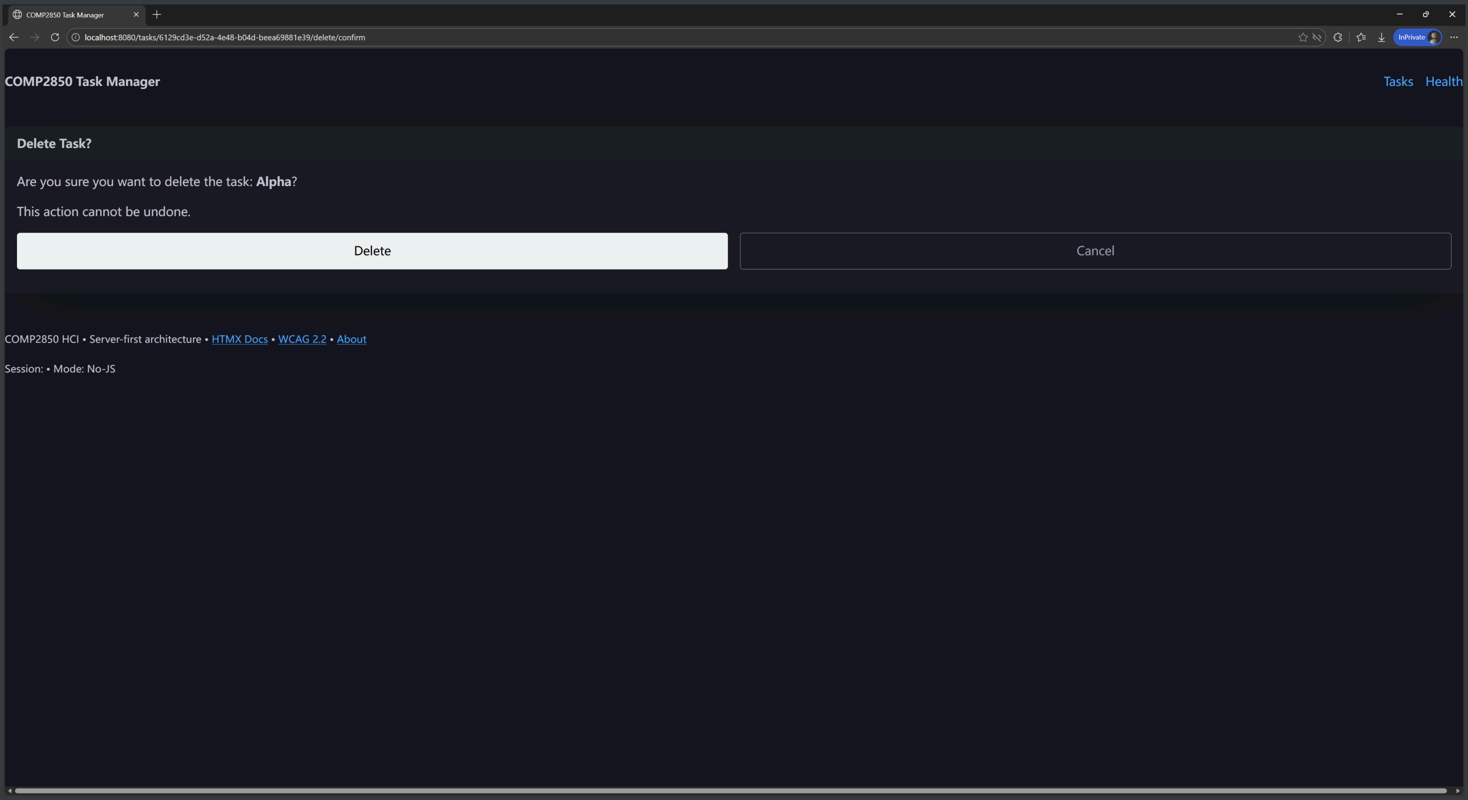
Based on the Inclusion-Severity Matrix, two issues were selected for immediate remediation:

1. **Priority 1 (Critical Safety)**: No-JS Delete Confirmation (Backlog ID 14).
2. **Priority 2 (Functional Blocker)**: Broken Edit Cancel Button (Backlog ID 15).

3.2 Fix 1: No-JS Delete Confirmation

Solution: We implemented an "Interceptor Route" pattern. For No-JS users, the "Delete" button now links to a dedicated confirmation page (`GET /tasks/{id}/delete/confirm`) instead of submitting a POST request immediately.

Implementation Evidence:


COMP2850 Task Manager
localhost:8080/tasks/6129cd3e-d52a-4e48-b04d-beea69881e39/delete/confirm
InPrivate

COMP2850 Task Manager

Tasks Health

Delete Task?

Are you sure you want to delete the task: **Alpha**?

This action cannot be undone.

Delete Cancel

COMP2850 HCI • Server-first architecture • HTMX Docs • WCAG 2.2 • About

Session: • Mode: No-JS

Figure 4: Safety gap closed via a dedicated confirmation route for No-JS users.

3.3 Fix 2: Robust Cancel Button

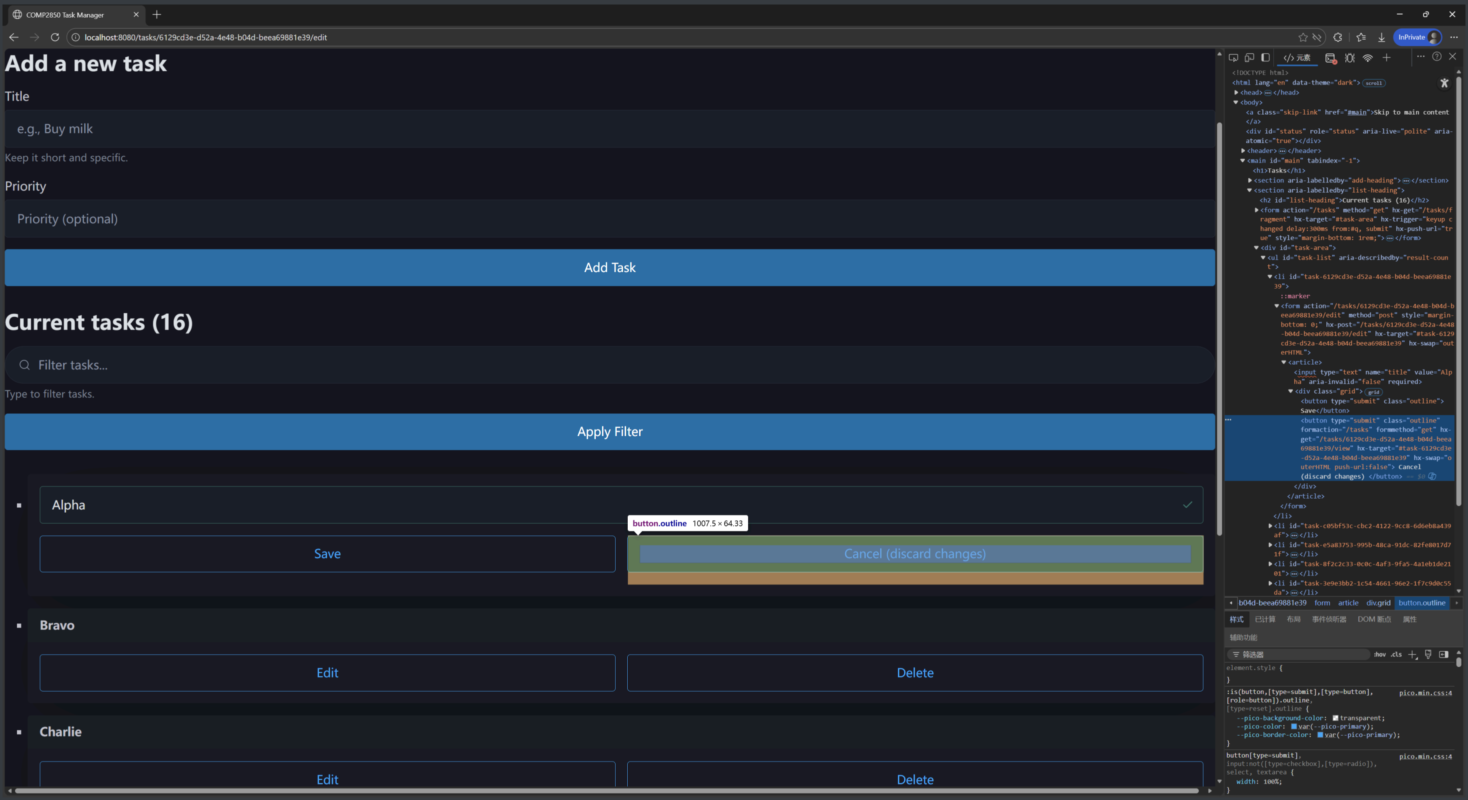
Solution: To fix the broken button while maintaining visual consistency with the "Save" button, we utilized the HTML5 `formaction` attribute.

Code Change:

```
    <button hx-get="...">Cancel</button>

    <button type="submit"
            formaction="/tasks"
            formmethod="get"
            hx-get="/tasks/{{ task.id }}/view"
            class="outline">
            Cancel
    </button>
```

Implementation Evidence:



```
<!DOCTYPE html>
<html lang="en" data-theme="dark"> scroll
  ▶<head> ⋯ </head>
  ▼<body>
     <a class="skip-link" href="#main">Skip to main content
     </a>
     <div id="status" role="status" aria-live="polite" aria-
     atomic="true"></div>
    ▶<header> ⋯ </header>
    ▼<main id="main" tabindex="-1">
       <h1>Tasks</h1>
      ▶<section aria-labelledby="add-heading"> ⋯ </section>
      ▼<section aria-labelledby="list-heading">
         <h2 id="list-heading">Current tasks (16)</h2>
        ▶<form action="/tasks" method="get" hx-get="/tasks/f
         ragment" hx-target="#task-area" hx-trigger="keyup c
         hanged delay:300ms from:#q, submit" hx-push-url="tr
         ue" style="margin-bottom: 1rem;"> ⋯ </form>
        ▼<div id="task-area">
          ▼<ul id="task-list" aria-describedby="result-coun
           t">
            ▼<li id="task-6129cd3e-d52a-4e48-b04d-beea69881e
             39">
               ::marker
              ▼<form action="/tasks/6129cd3e-d52a-4e48-b04d-b
               eea69881e39/edit" method="post" style="margin-
               bottom: 0;" hx-post="/tasks/6129cd3e-d52a-4e48
               -b04d-beea69881e39/edit" hx-target="#task-6129
               cd3e-d52a-4e48-b04d-beea69881e39" hx-swap="out
               erHTML">
                ▼<article>
                   <input type="text" name="title" value="Alp
                   ha" aria-invalid="false" required>
                  ▼<div class="grid"> grid
                     <button type="submit" class="outline">
                     Save</button>
                     <button type="submit" class="outline"
                     formaction="/tasks" formmethod="get" hx-
                     get="/tasks/6129cd3e-d52a-4e48-b04d-beea
                     69881e39/view" hx-target="#task-6129cd3e
                     -d52a-4e48-b04d-beea69881e39" hx-swap="o
                     uterHTML push-url:false"> Cancel
                     (discard changes) </button> == $0
                   </div>
                 </article>
               </form>
             </li>
            ▶<li id="task-c05bf53c-cbc2-4122-9cc8-6d6eb8a439
             af"> ⋯ </li>
            ▶<li id="task-e5a83753-995b-48ca-91dc-82fe8017d7
             1f"> ⋯ </li>
            ▶<li id="task-8f2c2c33-0c0c-4af3-9fa5-4a1eb1de21
             01"> ⋯ </li>
            ▶<li id="task-3e9e3bb2-1c54-4661-96e2-1f7c9d0c55
             da"> ⋯ </li>

b04d-beea69881e39  form  article  div.grid  button.outline

element.style {
}
:is(button,[type=submit],[type=button],          pico.min.css:4
[role=button]).outline,
[type=reset].outline {
   --pico-background-color: transparent;
   --pico-color: var(--pico-primary);
   --pico-border-color: var(--pico-primary);
}
button[type=submit],                             pico.min.css:4
input:not([type=checkbox],[type=radio]),
select, textarea {
   width: 100%;
}
```

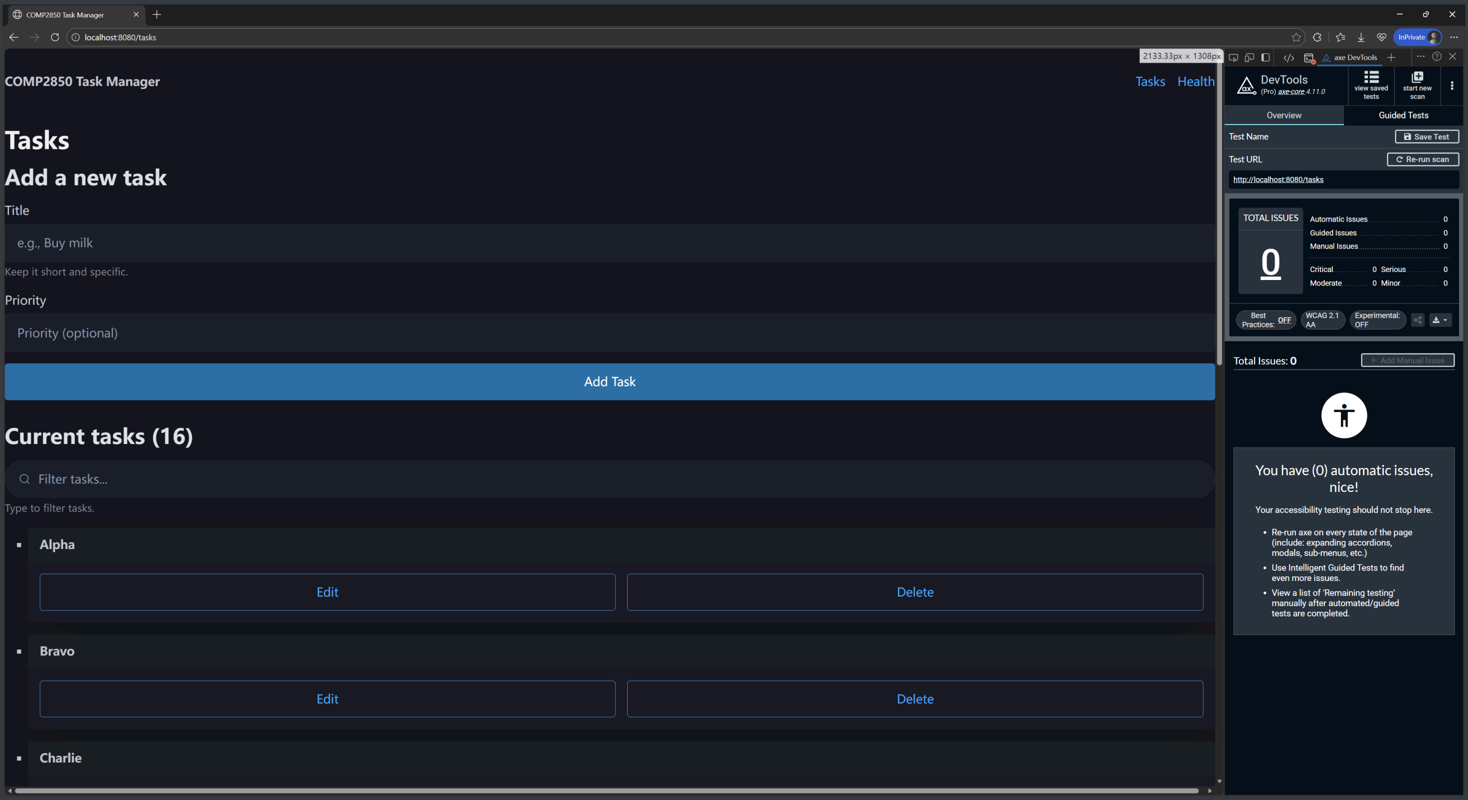
Figure 5: DevTools showing the formaction attribute, enabling No-JS navigation while keeping <button> styling.

Section 4: Verification & Reflection

4.1 Re-Verification

Manual regression testing confirmed that the fixes are effective:

- ☑ **No-JS**: Clicking "Delete" now leads to the confirmation page.
- ☑ **No-JS**: Clicking "Cancel" now correctly navigates back to the list view.
- ☑ **Automated Audit**: Axe tools confirmed no new accessibility violations were introduced.



COMP2850 Task Manager

Tasks Health

Tasks

Add a new task

Title

e.g., Buy milk

Keep it short and specific.

Priority

Priority (optional)

Add Task

Current tasks (16)

Filter tasks...

Type to filter tasks.

- Alpha
  Edit Delete
- Bravo
  Edit Delete
- Charlie

DevTools
(Pro) axe-core 4.11.0

view saved tests | start new scan

Overview | Guided Tests

Test Name — Save Test

Test URL — Re-run scan

http://localhost:8080/tasks

TOTAL ISSUES 0

Automatic Issues	0		
Guided Issues	0		
Manual Issues	0		
Critical	0	Serious	0
Moderate	0	Minor	0

Best Practices: OFF | WCAG 2.1 AA | Experimental: OFF

Total Issues: 0 — + Add Manual Issue

You have (0) automatic issues, nice!

Your accessibility testing should not stop here.

- Re-run axe on every state of the page (include: expanding accordions, modals, sub-menus, etc.)
- Use Intelligent Guided Tests to find even more issues.
- View a list of 'Remaining testing' manually after automated/guided tests are completed.

Figure 6: Automated accessibility audit confirming compliance.

4.2 Reflection

Process: The transition from evaluation to redesign highlighted the importance of "Progressive Enhancement".

Future Work: While safety and functionality are fixed, usability improvements like "Filter Persistence" (keeping search terms after reload) remain in the backlog for Semester 2.