

Key Contributions



Upvote / Agree System

Per-response agree button with live count; orange highlight on click



GitHub Migration & Org Setup

Migrated all team files to UserPostV2 repo; created walterArtifacts; guided teammates



Artifact Integration

Modularized all HTML artifacts into CSS/JS/HTML



Moderator Panel

Question submission with expiry timestamps; PUBLISH TO USER UI flow



Response Pipeline

Display of user responses + agree voting in othersResponses.html



Access Control

localStorage-based auth gate; admin token (WLH-admin-2026) → mod panel



Home Page

Login/create flow; dynamic content on auth state change



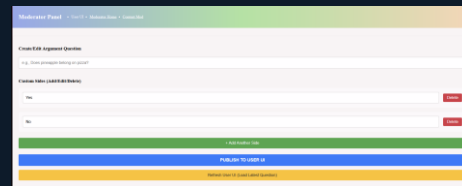
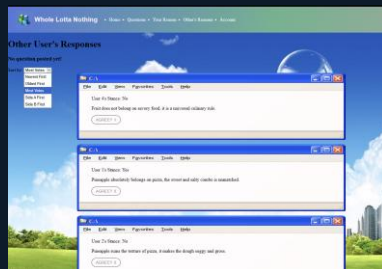
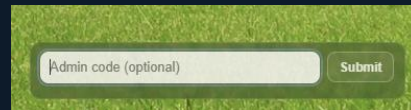
Sort by Upvotes

Dropdown sort on Other Responses page; persists through agree clicks



Navbar Redesign

Gradient logo, rounded links; full visual overhaul per client request



T-Shaped Skills · Evolution

SLIDE 2 OF 3

Walter A. · CS410 · Team 1

BROAD SKILLS (across all sprints)

Git / GitHub

Agile / Scrum

HTML & CSS

JavaScript

localStorage API

Team Integration

Navbar & UI Design

Client Comm.

BEFORE (Day 1)

- Basic HTML/CSS
- Intro JavaScript
- Git fundamentals
- Solo dev workflow

DEEP SKILLS

Frontend JS

Dynamic DOM, event handlers,
localStorage CRUD

System Integration

Modularizing teammates' code,
cross-page data flow

Feature Dev

Upvote logic, sort, auth gate,
admin token flow

UX / Responsiveness

Navbar redesign, site-wide redesign,
consistent styling

AFTER (End of Project)

- ✓ Full integration & modularization
- ✓ localStorage-based state management
- ✓ Scrum Master / team coordination
- ✓ Multi-page feature architecture
- ✓ Client-facing agile delivery

Reflections & Future Plans

SLIDE 3 OF 3

Walter A. · CS410 · Team 1

✓ What Went Well

Took on Scrum Master role; kept team unblocked with daily standups and sprint delegation

Successfully modularized & integrated all teammate artifacts

Delivered all client-requested features (auth, sort, mod panel, home page)

Strong code conventions and team dynamic - every feature was shipped following code conventions for more efficient integration

Discord channel improved async coordination across sprints

⚠ Challenges

Expiry timestamp integration broke question loading — required debugging

Custom cursor feature failed initially (browser security; idea ultimately scrapped)

→ Next Steps / Future Plans

Moderation System Completion

Finish flagging/reporting pipeline; moderator review already scaffolded

Rotating Question Queue

Client-requested queue of debate questions cycling on a timer

Broader Application

Patterns learned here apply directly to future full-stack internship work — REST APIs, auth middleware, modular component design