

# Site Audit Tool

Digitally Transforming the Warehouse Audit Process

## Project Brief

The Site Audit Tool is an interactive AWS based, interactive web interface for design engineers auditing delivery stations. Solving for the hurdles of the current audit process, the site audit tool improves audit efficiency, improves tracking of critical audit issues and helps engineers collaborate better with their team members.



Course

Graduate Capstone

IMT 597

Duration



7 months

Nov '23 - May '24

Sponsor



David Leonardo

Design Engineer, AMZL

## Background

Amazon's Last Mile team manages the final stage of package delivery, from the delivery station to the customer's doorstep. As Amazon grows its delivery station network, sites with different capacities are opened every few months. Before launching, Last Mile Design Engineers audit each site to ensure that its interior and exterior match the planned designs. Any discrepancies or issues are recorded in an audit template for further review.

## User Research

We interviewed four design engineers from the Last Mile team to better understand the status quo; the current audit process, the challenges it presents and how the team overcomes these challenges currently.

A major bottleneck, branching out to other challenges was the spreadsheet based **Site Audit template**.

### Problem Areas Identified



Collaboration

Relying on the static audit template and emails for communicating audit observations led to inefficient collaboration between auditors.



Interactiveness

Auditors reported that the static nature of the current spreadsheet template is limiting and unintuitive.



Tracking Progress

The current audit process poses challenges in tracking the progress of issues raised and changes made.



Integration

The applications used during audits work in silos and are not integrated for easier access and use.

### Problem Statement

The site audit process is limited by a spreadsheet based audit template and lacks seamless information channels, causing functional bottlenecks, information errors, in turn leading to audit errors and incorrect reporting.

## Methodology

The dummy site audit template and dummy audit data, provided to us by the sponsor, were key components to the solution formulation. Coupled with the capabilities of AWS, we simultaneously initiated the building UX framework and the backend architecture.



- Identified problem categories from interview data through "Affinity mapping"
- Created the information architecture of the web interface
- Wire-framing. Review. Reiterate.
- Developed high fidelity prototype screens from the feedback received.



- Set up Django with React to enhance backend functionality.
- Integrated AWS services (S3, RDS, Lambda) for improved performance and reliability.
- Developed and tested APIs for Site Audit and Discrepancy functionalities, enabling efficient data interactions via GET and POST methods.
- Implemented real-time data handling and output mechanisms (PDF generation and logging features)



- Developed UI facilitating seamless switching between audit modules to streamline workflow.
- Integrated dynamic modal pop-ups for initiating audits, improving data entry and user engagement.
- Enhanced audit tables with interactive elements (match status icons and real-time validation) to boost data accuracy and user involvement.
- Improved integration with backend services for real-time data sync and improved performance.

## Final Solution

Here's a quick break down of our final solution, with the user experience and the technical specification highlights. We prioritized function over form and kept UI consistent with Amazon Logistics' website.

### Development </>

#### Data Architecture & Integration

We employed **Django and React** to create a robust backend framework. **RESTful APIs** have been implemented to facilitate smooth and responsive data flow.

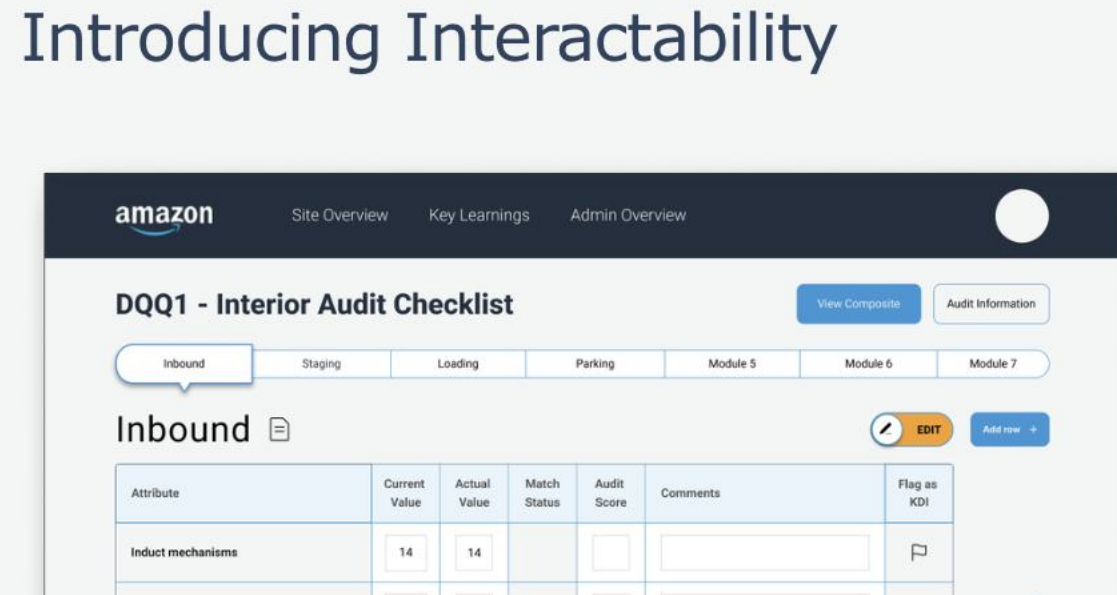


#### Cloud Services and Scalability

AWS services such as **S3, RDS, and Lambda** have been fully integrated, enhancing the system's scalability and reliability, effectively supporting high-demand operations.

### User Experience </>

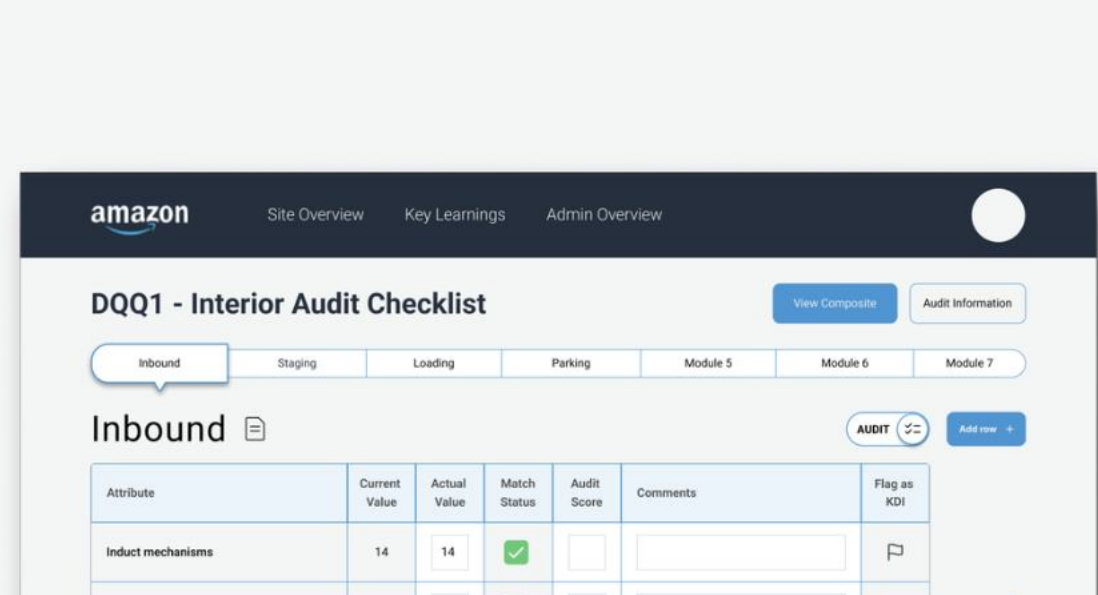
#### Introducing Interactability



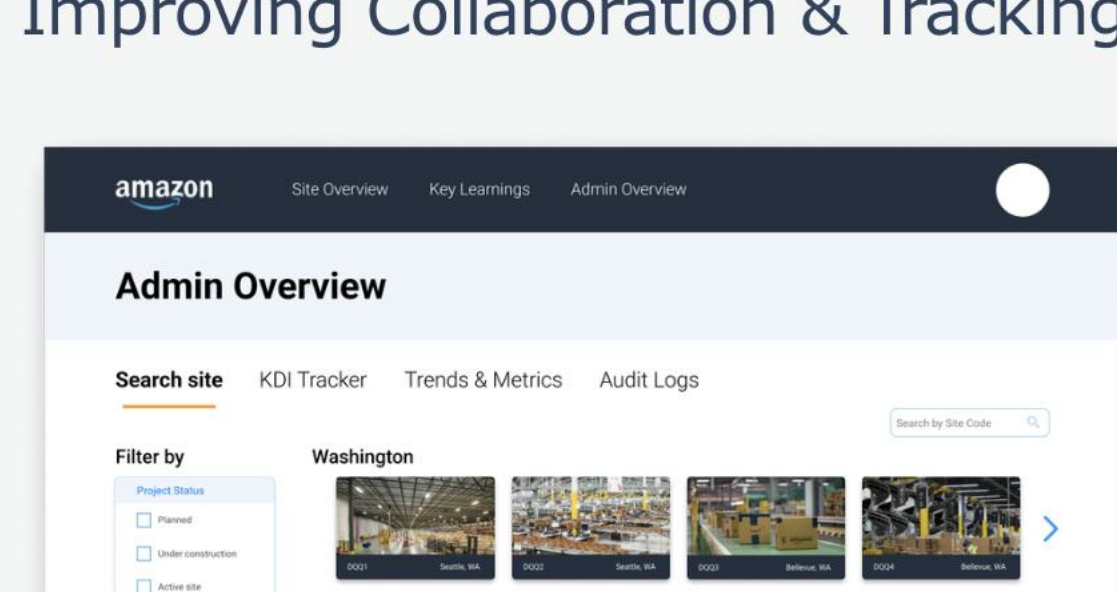
The **Module bar** lets auditors toggle between different site modules and tracks progress.

The **"Audit - Edit"** switch allows for quick, on the spot changes for non-critical issues.

Key Design Issues (KDI) are **flagged & tracked** separately.

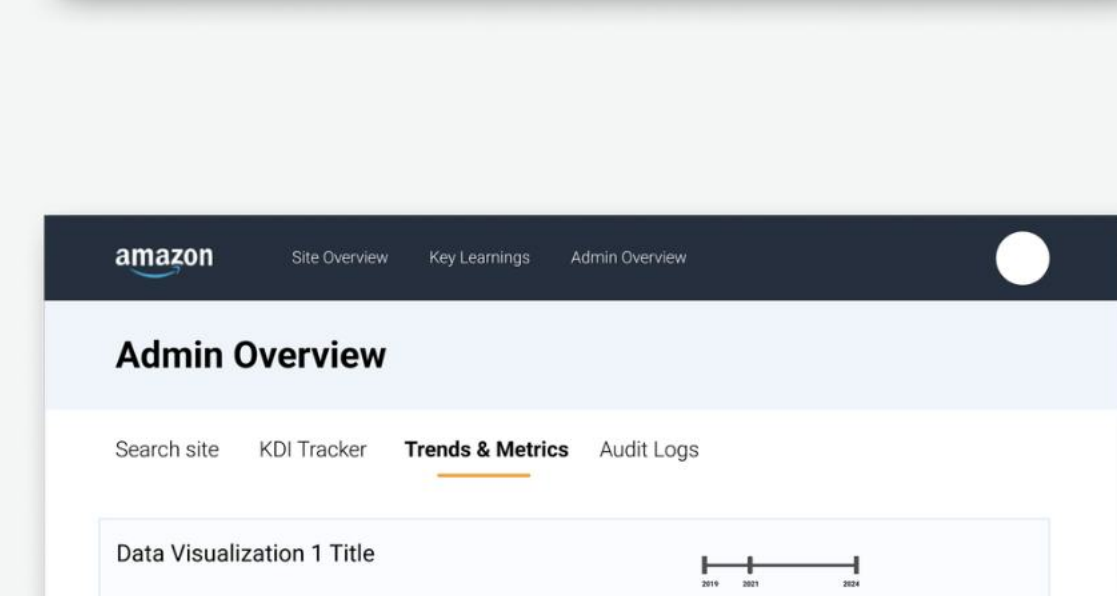
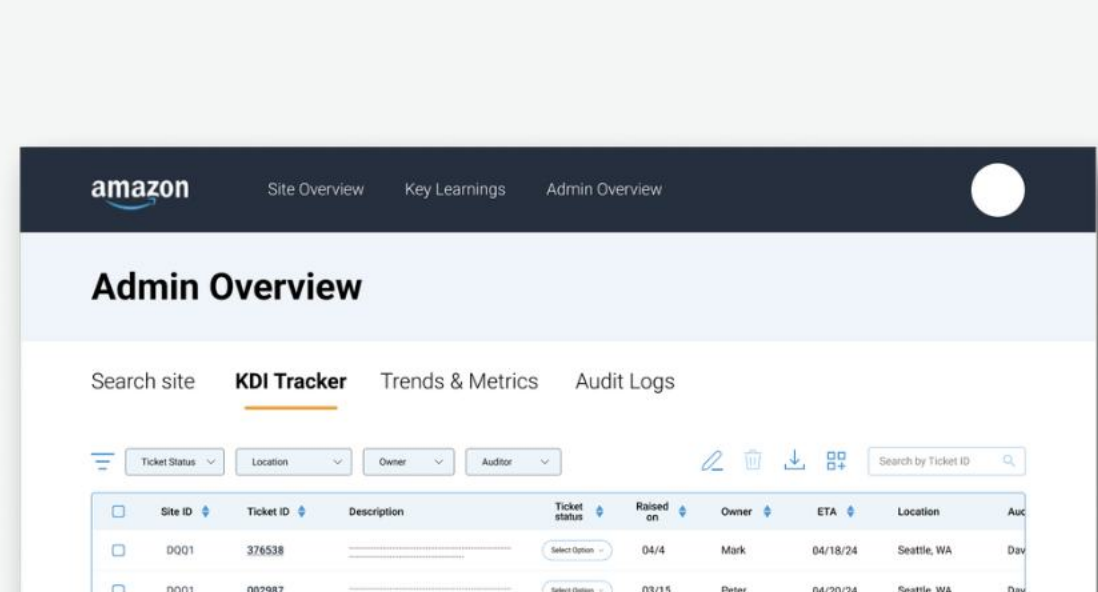


#### Improving Collaboration & Tracking



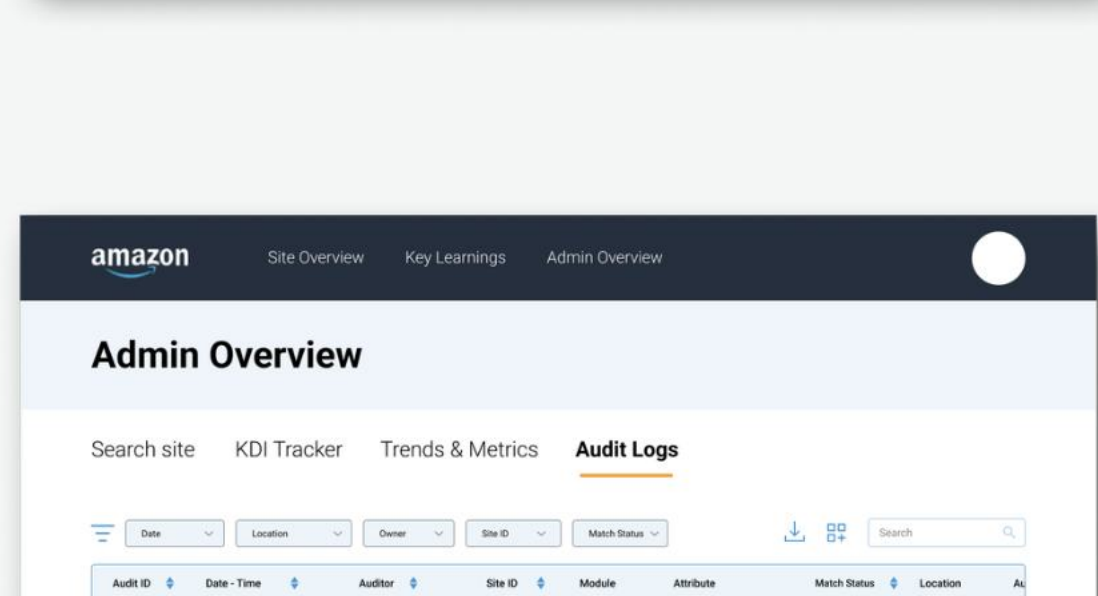
An admin overview page allows auditors to **search specific sites** for status updates on projects at a site.

The **Key Design Issue (KDI) Tracker** helps monitor critical audit issues.



**Dashboards and charts** give quick summaries of projects, project and audit metrics.

The **Audit Log** - A running register of all audit changes for easy retroactive review.



## Projected Benefits & Impact



**Highlighting KDIs:** By introducing separate tracking mechanisms for Key Design Issues, the visibility of critical-to-launch issues is increased.



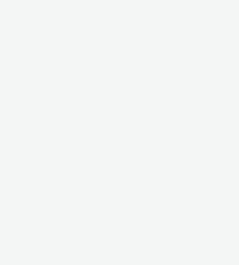
**Accelerated Efficiency:** Saves 1 hour per audit by rapidly identifying and reporting discrepancies.



**Dynamic Auditing Interface:** Adapts to real-time site attributes, ensuring each audit is precise and tailored.



**Boosts Collaboration:** Enhances teamwork between field auditors and engineers, minimizing re-audits.



**Double the Detection:** Identifies 100% more issues monthly, accelerating problem resolution.

### Team Phoenix



Sanyam



Siddharth



Stushi



Ashwin

