Accuknox Assignment by Prashant Yadav

Problem Statement 1:
Title: Product Requirement and Low-Fidelity Wireframes

Product Requirements Document

1. Objective

   The objective is to build a security tool that scans container images for known vulnerabilities and provides actionable insights to users. The tool will help users identify critical or high vulnerabilities within their container images, prioritize which images to fix, and streamline the remediation process.

2. User Personas

   - **DevOps Engineers:** Need to ensure container images are secure before deploying them into production.
   - **Security Analysts:** Require detailed insights into vulnerabilities for proactive risk management.
   - **Developers:** Want to quickly identify and fix vulnerabilities within their container images.

3. User Stories

   - **As a DevOps Engineer,** I need to view a list of all container images and their associated vulnerabilities so that I can assess the security risk of each image.
   - **As a Security Analyst,** I need to filter images by the severity of vulnerabilities (Critical, High, Medium, Low) to prioritize which images require immediate attention.

- **As a Developer,** I need to drill down into specific vulnerabilities to understand their impact and remediation steps so that I can fix the issues promptly.
- **As a user,** I need to understand which container images have vulnerabilities and how severe they are.
  If there are any critical or high vulnerabilities, I need to address them by identifying which images require fixing. I have thousands of images in my repository. Help us create a product requirements document or wireframe that can assist users in solving these issues.

4. Key Features

- **Image Dashboard:** Displays a list of container images with high-level details about the number of vulnerabilities and their severity.
- **Vulnerability Filtering:** Allows users to filter images based on vulnerability severity (e.g., Critical, High, Medium, Low).
- **Detailed View:** Provides a detailed breakdown of vulnerabilities for a selected container image, including description, severity, and remediation guidance.
- **Search and Sort Functionality:** Users can search and sort images based on different parameters (e.g., name, number of vulnerabilities, severity).
- **Bulk Actions:** Users can select multiple images and initiate actions like "re-scan" or "mark as reviewed."
- **Notifications:** Alerts users when new vulnerabilities are detected in existing images.

5. Non Functional Requirements

- **Performance:** The tool should be able to scan thousands of images without significant delay.
- **Scalability:** The system should support growing image repositories as user needs expand.

- **Security:** Secure access with role-based permissions to ensure only authorized users can view or manage vulnerabilities.

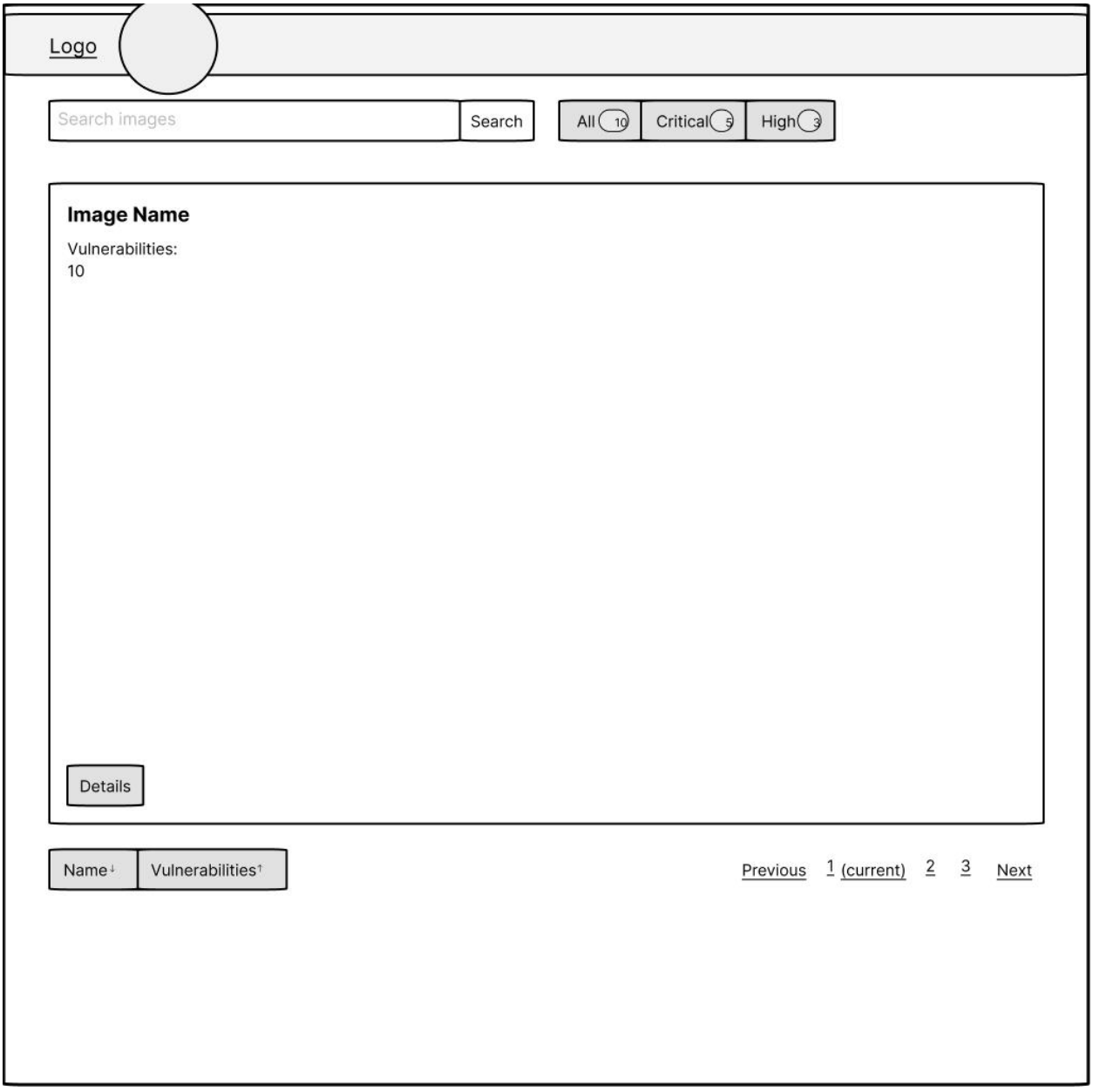
6. UI Requirements
    - The UI should be intuitive and easy to navigate.
    - Critical and high vulnerabilities should be visually highlighted.
    - The design should support both light and dark modes for user preference

7. Development Actions (Optional Task)
    - **Backend:** Develop API endpoints for scanning images, retrieving scan results, and filtering vulnerabilities.
    - **Frontend:** Build the UI for the image dashboard, filtering options, and detailed vulnerability views.
    - **Database:** Set up a database to store scan results, images metadata, and vulnerability details.
    - **Security:** Implement authentication and authorization for secure access to the tool.

Low-Fidelity Wireframes

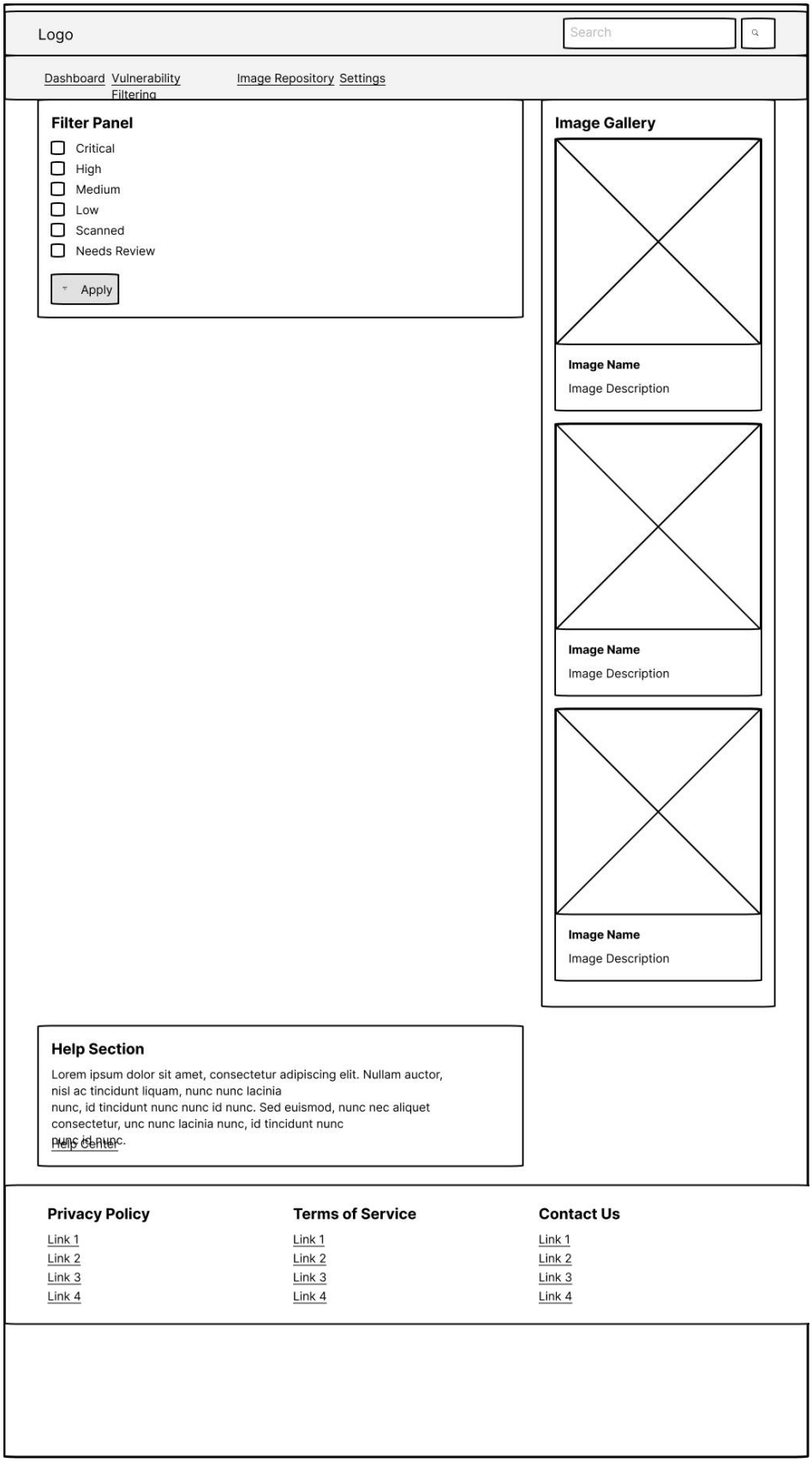
1. **Image Dashboard**
    - **Purpose:** Shows a list of container images with their vulnerability counts and severity.
    - **Components:**
        - **Header:** Search bar, filter options, and profile settings.
        - **Image List:** Displays image name, vulnerability counts, and severity distribution with visual indicators.
        - **Sort & Pagination:** Options to sort images and navigate through pages.

- **Helps Users:** Quickly identify images with critical vulnerabilities and prioritize fixes.



2. Vulnerability Filtering

- **Purpose:** Allows users to filter images based on vulnerability severity and status.
- **Components:**
  - **Filter Panel:** Checkboxes for severity (Critical, High, Medium, Low) and status filters (Scanned, Needs Review).
  - **Apply Button:** Updates dashboard with selected filters.
- **Helps Users:** Focus on the most urgent vulnerabilities and manage large image repositories efficiently.


Logo
Search
Dashboard
Vulnerability Filtering
Image Repository
Settings
Filter Panel
Critical
High
Medium
Low
Scanned
Needs Review
Apply
Image Gallery
Image Name
Image Description
Image Name
Image Description
Image Name
Image Description
Help Section
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam auctor, nisl ac tincidunt liquam, nunc nunc lacinia
nunc, id tincidunt nunc nunc id nunc. Sed euismod, nunc nec aliquet consectetur, unc nunc lacinia nunc, id tincidunt nunc
nunc id nunc.
Help Center
Privacy Policy
Link 1
Link 2
Link 3
Link 4
Terms of Service
Link 1
Link 2
Link 3
Link 4
Contact Us
Link 1
Link 2
Link 3
Link 4

3. **Detailed View**

- **Purpose:** Provides an in-depth view of vulnerabilities for a selected image.
- **Components:**
  - **Header:** Displays image name, scan date, and summary.
  - **Vulnerability List:** Detailed rows with ID, description, severity, impact, and remediation steps.
  - **Action Buttons:** Options like "Mark as Fixed" or "Re-scan."
- **Helps Users:** Understand specific vulnerabilities and take immediate action to resolve them.

Image Vulnerability Scanner

Scan Date: [ScanDate]

Summary: [Summary]

Re-scan

Vulnerability List

ID	Description	Severity	Impact	Remediation Steps	
1	[Description]	[Severity]	[Impact]	[Remediation Steps]	Mark as Fixed
2	[Description]	[Severity]	[Impact]	[Remediation Steps]	Mark as Fixed

Charts

Help Section

[Help Section Content]

Submitted By - Prashant Yadav

Email - prashantyadav358@gmail.com Phone

No. - 9992236415