1. Product Requirements Document:

product requirements would include:

User Stories:

As a user, I want to see a list of container images with vulnerabilities.

As a user, I need to know the severity (low, medium, high, critical) of the vulnerabilities.

As a user, I want to filter container images based on vulnerability severity.

As a user, I want to prioritize images that need to be fixed urgently (e.g., high/critical vulnerabilities).

As a user, I want to easily navigate and search through thousands of images to find those that require immediate attention.

Vulnerability Dashboard: A central interface showing a list of scanned container images and their associated vulnerabilities.

Severity Indicators: Color-coded tags or icons indicating vulnerability severity (critical, high, medium, low).

Filtering & Sorting: Options for filtering images by severity and sorting them by vulnerability count.

Image Details Page: Clicking on a container image opens a detailed view showing the vulnerability specifics, affected dependencies, and potential fixes.

Bulk Actions: Tools that allow users to take actions on multiple images (e.g., marking for fix).

Search & Analytics : Search capabilities across all images and vulnerabilities, possibly with basic analytics (like most common vulnerabilities).

2. Low-Fidelity Wireframes:

The wireframes should represent the structure and layout of the interface, focusing on functionality rather than aesthetics. Here’s how the UI might be structured:

Dashboard View:

A table or list of container images, each row showing:

Container name

Number of vulnerabilities (categorized by severity)

Last scanned date

Filters for vulnerability severity and search functionality on top.

Detail View (for a specific container image):

Image-specific details such as:

List of vulnerabilities

Affected components

Severity for each vulnerability

Links to suggested fixes or remediation steps

The wireframe could be as simple as hand-drawn sketches, using placeholders for where text and icons would go.

3. Bonus - Development Action Items:

Frontend: Build the UI based on the wireframe, ensuring users can filter and sort vulnerabilities.