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

**Product Requirements Document (PRD)**

**1. Overview**

**Title: Container Image Security Scanner**

**Background:**

A security product is required to scan container images, detect vulnerabilities, and provide insights into the severity of these vulnerabilities. Users need to identify which container images require immediate attention and remediation.

**Target Audience:**

* DevOps Engineers
* Security Analysts
* Platform Engineers
* IT Administrators

**2. Objectives**

* Provide users with visibility into vulnerabilities in their container images.
* Categorize vulnerabilities by severity (Critical, High, Medium, Low).
* Offer remediation guidance to fix critical and high vulnerabilities.
* Enable filtering and sorting of images based on vulnerability status.
* Support bulk actions for efficient vulnerability management.

**3. Features & Requirements**

**3.1. Image Scanning & Vulnerability Detection**

* Automatically scan container images for vulnerabilities.
* Display CVE (Common Vulnerabilities and Exposures) details.
* Show affected components and package versions.

**3.2. Severity Classification**

* Categorize vulnerabilities as Critical, High, Medium, or Low.
* Highlight images with critical and high vulnerabilities prominently.

**3.3. Search, Filter & Sort**

* Search images by name or repository.
* Filter by severity level, status (fixed/unfixed), and last scan date.
* Sort images based on the number of vulnerabilities or severity level.

**3.4. Remediation Recommendations**

* Provide recommended fixes (e.g., update to a secure package version).
* Link to official security advisories for further reference.

**3.5. Bulk Actions & Notifications**

* Enable bulk selection for marking images for remediation.
* Send alerts for newly detected critical vulnerabilities.

**4. Wireframes (Low-Fidelity Sketches)**

**4.1. Dashboard View**

* Overview of scanned images with vulnerability statistics.
* A heatmap representation of vulnerabilities across repositories.

**4.2. Image Details Page**

* List of vulnerabilities with severity indicators.
* Expandable sections for detailed CVE information.

**4.3. Filters & Actions Panel**

* Dropdown for severity filters.
* Bulk selection checkbox with 'Mark for Remediation' button.

(Include simple wireframe sketches here)

**5. Development Action Items (Optional Bonus)**

**5.1. Backend Development**

* Implement container image scanning using an open-source scanner (e.g., Trivy, Clair).
* Integrate CVE database for vulnerability detection.
* Develop APIs to fetch image scan results and metadata.

**5.2. Frontend Development**

* Create a dashboard UI with real-time vulnerability updates.
* Implement filtering and sorting features.
* Design a detailed vulnerability page with remediation suggestions.

**5.3. DevOps & Deployment**

* Automate periodic scanning via CI/CD pipeline.
* Set up notifications for critical vulnerabilities via email/slack.

**6. Conclusion**

This product aims to streamline vulnerability detection and remediation for container images. By providing a clear and actionable interface, users can effectively manage security risks in their container repositories.

**Wireframe for your Container Image Vulnerability Scanner**

