1. Product Requirement and Low-Fidelity Wireframes

Executive Summary

Modern organizations often work with thousands of container images. These container images may have known or unknown security vulnerabilities that, if not detected and addressed early, can lead to production failures or security breaches. This product aims to help users:

- Identify which images have vulnerabilities.
- Understand the severity (Critical, High, Medium, Low).
- Prioritize fixing Critical/High severity issues.
- Take immediate action through a user-friendly dashboard.

Target Users

User Role	Responsibilities
DevOps Engineers	Integrate scans into CI/CD; fix vulnerabilities.
Security Engineers	Monitor risk posture and compliance.
Developers	Fix issues in dependencies and base images.
System Admins	Ensure only secure images are deployed.
Compliance Teams	Ensure security standards are met.

Use Cases

1. View and filter thousands of container images.
2. Identify images with critical/high vulnerabilities.
3. Deep dive into vulnerabilities with CVE details.
4. Rescan or reverify fixed images.
5. Configure alerts for new/critical vulnerabilities.

User Stories

- “As a user, I want to integrate image scanning into CI/CD so I can catch vulnerabilities early.”
- “As a user, I want to see an overview of all vulnerable images with severity breakdowns.”
- “As a user, I want to rescan fixed images to confirm issues are resolved.”
- “As an admin, I want to get alerts when critical vulnerabilities are found.”

Key Features

Feature	Description
Image Vulnerability Scan	Scan images using tools like Trivy or Clair.
Dashboard View	High-level overview of vulnerabilities.
Vulnerability Detail Page	CVE ID, severity, affected packages, fix versions.
CI/CD Integration	Scans triggered during builds or pushes.
Alert System	Configurable alerts for new or critical findings.
API Access	REST APIs for integrations and automation.

Low-Fidelity Wireframes(visual attached in the end)

Dashboard Screen

- Search bar (image name)
- Filters: Severity (Critical, High, Medium, Low)
- Summary section: Total images scanned, Critical vulnerabilities count, High vulnerabilities count
- Table: Image Name, Repository, Vulnerability Count (with color indicators), Last Scan Date, Actions (Rescan, View Details)

Image Details Page

- Image metadata: name, repo, scan date
- Vulnerability table: CVE ID, Severity, Affected Package, Installed Version, Fixed Version, Action

Alerts Configuration Page

- Set thresholds (e.g., Critical only)
- Notification method (Email, Slack)
- Add recipients
- Toggle: Real-time / Batched alerts
- Save Settings button

Development Action Items

Module	Task
Vulnerability Scanner	Integrate tools like Trivy or Grype. Schedule periodic/trigger-based scans.
CI/CD Integration	Create hooks for GitHub Actions, Jenkins, GitLab, etc.
API Design	REST APIs for triggering scans, fetching results, bulk actions.
Prioritization Logic	Sort/filter vulnerabilities based on severity and frequency.

Frontend UI	Build dashboards using React/Angular with filtering and charts.
Notification Service	Send alerts via email or Slack; admin-controlled settings.
Bulk Actions	Allow multi-image scanning or fixing.
Role-Based Access	Define roles like Admin, Viewer, Developer.

Success Metrics

- Reduction in average critical vulnerabilities per image.
- Faster remediation time (mean time to fix).
- Reduction in post-deployment vulnerability incidents.
- Improved compliance coverage.

Low Fidelity Wireframe visual