Product Requirements for Container Image Vulnerability Scanner

**Objective**

Develop a product that scans container images for known vulnerabilities and displays findings to users, including severity levels and recommended actions for remediation.

**Problem statement**

Users have thousands of container images, each potentially containing known vulnerabilities that can compromise application security. Users need a way to efficiently scan, identify, and remediate these vulnerabilities.

**Functional Requirements**

1. **User Authentication**

* Users must authenticate before accessing the scanning results.
* Ensures that only authorized personnel can access sensitive vulnerability data.

1. **Image Scanning**

* The system scans container images for vulnerabilities from different repositories.
* Provides a detailed understanding of the security posture of each image.
* **Flow**:
  + - 1. User selects the images they want to scan.
      2. The system runs a scan and retrieves vulnerability data.
      3. Results are displayed in the user interface.

1. **Vulnerability Reporting**

* Displays a list of vulnerabilities identified in each image.
* Allows users to see which images need attention and the severity of issues.
* Implementation Details:
  + Vulnerabilities are categorized as Critical, High, Medium, or Low.
  + Include details like ID, description, and severity score.
  + Sorting and filtering options based on severity, date discovered, and image.
  + Users can click on a vulnerability to see detailed information.
* Example of the Output:

|  |  |  |  |
| --- | --- | --- | --- |
| **Image Name** | **Vulnerability** | **Severity** | **Fix Available** |
| nginx:1.14 | CVE-2022-1234 | Critical | Yes |

1. **Remediation Guidance**

* Provides actionable steps to fix critical and high-severity vulnerabilities.
* Assists users in resolving vulnerabilities promptly.
* Implementation Details:
  + Recommendations are sourced from official security advisories.
  + Display upgrade paths or patches available for affected components.
* Example:
  + For CVE-2022-1234, the system suggests upgrading to nginx version 1.16.

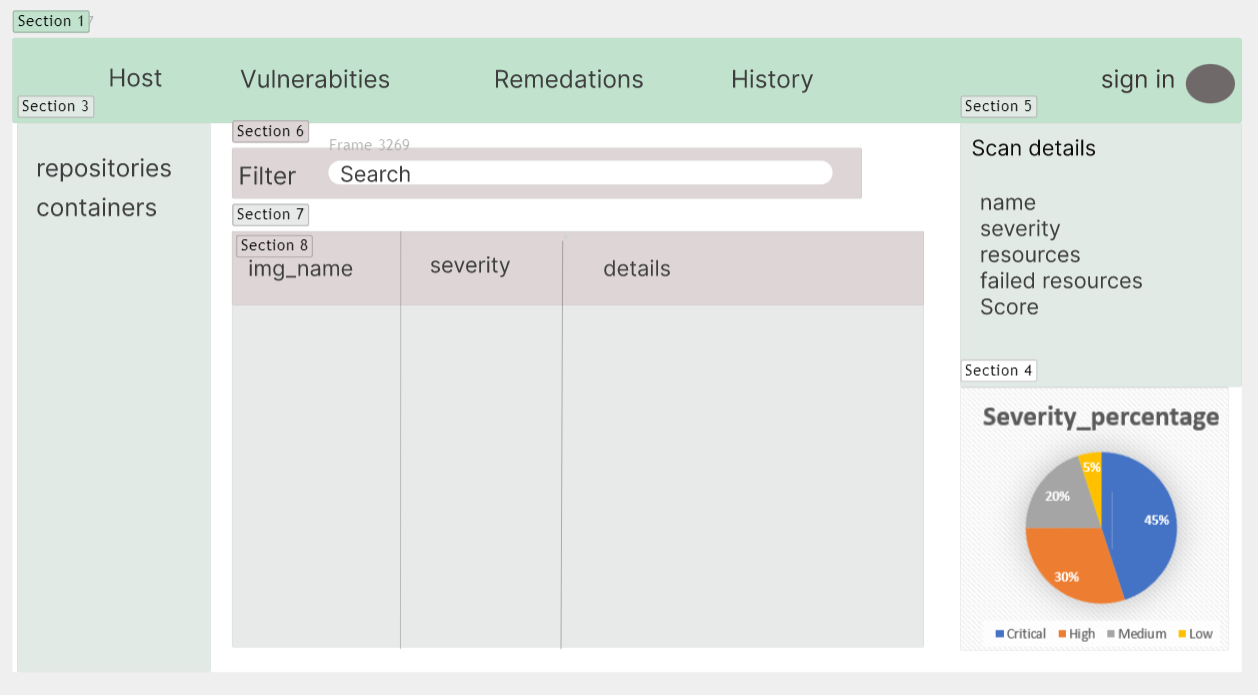
1. **Dashboard and Reporting**

* Overview of vulnerabilities across all scanned images.
* Provides a quick summary of the security state of all images.
* Implementation Details:
  + A pie chart showing vulnerability distribution by severity.
  + List of top 5 most vulnerable images.
* Example:

**Development Action Items**

|  |  |  |
| --- | --- | --- |
| Development actions | Description | |
| **Scanning Infrastructure** | | Deploy containerized scanners using Kubernetes. |
| **UI/UX Development** | | Develop wireframes and mockups.  Build front-end components using React or Angular. |
| **Testing** | | Unit testing for the scanning modules.  Load testing for performance under high image counts. |
| **Documentation** | | User guides and API documentation. |

**low-fidelity wireframes for the user interface**

****