Problem Statement 1: Product Requirement and LowFidelity Wireframes

Background/Task: Users need to scan container images to find vulnerabilities. The product must help users understand which container images are vulnerable and the severity of those vulnerabilities.

Solution:

1. Product Requirements Document:

User Interface: Provide a dashboard showing a list of scanned container images with the number and severity of vulnerabilities (Critical, High, Medium, Low).

Sorting/Filtering: Enable sorting and filtering based on the severity, age of the vulnerability, and image tags.

Detailed View: For each image, show the specific vulnerabilities, recommended fixes, and links to external CVE (Common Vulnerabilities and Exposures) reports.

Alerts: Configure automated alerts for critical vulnerabilities.

Search: Add search functionality to allow users to quickly find specific images.

2. LowFidelity Wireframes: Create simple sketches showing:

A landing page with the dashboard for scanned images.

Imagespecific vulnerability details, sortable columns, and buttons for filtering severity.

3. Bonus:

Development action items could include setting up API calls to a vulnerability database (like CVE) and integrating with container registries.

Problem Statement 2: Kubernetes Security Scan

Background/Task: Install a local Kubernetes cluster and scan it using a tool like Kubescape to detect security vulnerabilities.

Solution:

Install Minikube (or any other local K8s cluster) and Kubescape.

Use Kubescape to scan for vulnerabilities and security misconfigurations in the Kubernetes cluster.

Deliver the JSON file containing the findings from the scan.

Problem Statement 3: Technical Challenge

Step 1:

Create a GoLang program that reflects the current date and time.

Host this GoLang program on GitHub.

Use Docker to containerize the GoLang program and push the container image to DockerHub.

Step 2:

Deploy the GoLang container to Kubernetes using a declarative YAML file with 2 replicas.

Step 3:

Expose the GoLang web application to the internet using Kubernetes Service or Ingress.

Use resources like Qwiklabs or your own cloud platform (GCP) for testing and exposing the application.