Product Manager Task Submission

1. Product Requirements Document (PRD)

Product Requirements Document (PRD)

Product Name: Container Vulnerability Dashboard (CVD)

Overview:

A tool to scan container images for vulnerabilities and show them in a web-based dashboard.

Key Features:

- Image listing with vulnerabilities count
- Filter by severity
- View detailed CVEs per image
- Fix suggestions
- Export reports

User Stories:

1. View all images with vulnerability info
2. Filter by severity levels
3. Drill down to CVE details
4. Mark as resolved/ignored
5. Export to CSV/PDF

Success Metrics:

- Time to detect and resolve critical issues
- % of resolved CVEs within SLA

2. Kubernetes Security Scan Results (Mock)


```
{
  "cluster": "minikube",
  "results": [
    {
```

```
      "control": "Networking",
      "score": 75,
      "status": "Warning"
    },
    {
      "control": "RBAC",
      "score": 50,
      "status": "Fail"
    },
    {
      "control": "Secrets",
      "score": 90,
      "status": "Pass"
    }
  ]
}
```

3. GoLang Web App + Docker + Kubernetes

main.go:

```
package main

import (
    "fmt"
    "net/http"
    "time"
)

func handler(w http.ResponseWriter, r *http.Request) {
    fmt.Fprintf(w, "Current date and time: %s", time.Now().Format(time.RFC1123))
}

func main() {
    http.HandleFunc("/", handler)
    http.ListenAndServe(":8080", nil)
}
```

Dockerfile:

```
FROM golang:1.20-alpine
WORKDIR /app
COPY . .
RUN go build -o datetime-app
CMD ["./datetime-app"]
```

deployment.yaml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: datetime-app
```

```
spec:
  replicas: 2
  selector:
    matchLabels:
      app: datetime
  template:
    metadata:
      labels:
        app: datetime
    spec:
      containers:
      - name: datetime
        image: your-dockerhub-username/datetime-app:latest
        ports:
        - containerPort: 8080
```

service.yaml:

```
apiVersion: v1
kind: Service
metadata:
  name: datetime-service
spec:
  type: LoadBalancer
  selector:
    app: datetime
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8080
```