🔒 Security Report: 3-Tier Node.js ToDo Application

Candidate: Priyansh Kumar Pandey
Role: DevSecOps Candidate Evaluation
Date: July 2025

📄 Project Overview

This project demonstrates a secure, containerized deployment of an advanced ToDo list application using a 3-tier architecture:

- **Frontend:** HTML + JavaScript (served via Nginx)
- **Backend:** Node.js + Express
- **Database:** MongoDB (containerized)

The stack is fully Dockerized with a multi-stage build and deployed using Docker Compose. CI/CD is managed via GitHub Actions.

🛡️ Security Measures Implemented

🔗 1. Minimal Base Image

- Used `node:18-alpine` for a slim, secure Node.js environment.
- Nginx is also based on `alpine` to reduce attack surface.

🔗 2. Multi-Stage Docker Build

- Builds backend and frontend separately, then merges into a final runtime image.
- Reduces image size and removes dev-time dependencies.

🔗 3. Trivy Image Scan (CI)

- GitHub Actions runs `Trivy` on every push to `master`.
- CI fails if any **HIGH** or **CRITICAL** vulnerabilities are detected.
- CVEs like `CVE-2024-21538` were mitigated by upgrading `cross-spawn`.

🔗 4. Non-Root User in Dockerfile

- Final Docker image runs the app as the non-root `node` user.
- Prevents privilege escalation vulnerabilities.

🔗 5. Secrets Management

- Application uses `.env` file (gitignored).

- CI/CD secrets (Docker credentials) stored securely in GitHub Secrets.

🔗6. Docker Compose Deployment

- All containers isolated via user-defined bridge network.
- MongoDB volume persisted securely.

🔗7. CI/CD Pipeline (GitHub Actions)

- Builds and scans the image.
- Pushes to Docker Hub **only** if Trivy scan passes.
- Includes CI status badge in `README.md`.

📒Additional Security Recommendations

Area	Recommendation
Static Code Analysis	Integrate Semgrep in CI to detect code smells and insecure patterns
Secrets Management	Use AWS Secrets Manager or Vault for production secret injection
Runtime Security	Use Docker `seccomp` or AppArmor profiles, or Falco for anomaly detection
HTTPS	Terminate SSL using Nginx or behind a cloud load balancer

---

🔧Conclusion

This project demonstrates DevSecOps best practices including secure image builds, vulnerability scanning, CI/CD gating, and least privilege Docker containers. With further enhancements like Semgrep and runtime security tools, this app is ready for staging or production.

Submitted by:
Priyansh Kumar Pandey
GitHub Repo
Docker Hub