# Task 2 Documentation: Vue.js Application Deployment Using Docker

## 1. Objective

The objective of Task 2 is to containerize and deploy a Vue.js frontend application using Docker by following enterprise-level DevOps practices. This ensures consistency, scalability, and production readiness.

## 2. Scope

This document explains the end-to-end deployment of a Vue.js application using Docker, focusing on industry standards followed by MNCs for frontend application delivery.

## 3. Architecture Overview

1. Vue.js frontend application
2. Node.js for build process
3. Multi-stage Docker build
4. Nginx for serving static content

## 4. Project Structure

1. src – Vue.js source code
2. public – Static assets
3. Dockerfile – Build and runtime definition
4. package.json – Dependencies and scripts
5. nginx.conf – Nginx configuration

## 5. Docker Strategy

A multi-stage Dockerfile is used to build the Vue.js application and serve the optimized production build using Nginx. This reduces image size and improves security.

## 6. Implementation Steps

1. Build Vue.js application using Node.js
2. Create Docker image using multi-stage build
3. Run container with proper port mapping
4. Access application via browser

## 7. Security Considerations

1. No sensitive data inside Docker image
2. Minimal base images
3. Static file serving via Nginx

## 8. Verification

The application was verified by accessing it through the Docker-exposed port, confirming successful containerization and deployment.

## 9. Enterprise Best Practices

1. Docker-based deployment

2 Optimized production build
3 Scalable frontend architecture
4 Cloud-ready deployment

## 10. Conclusion

The Vue.js application was successfully deployed using Docker following MNC-level DevOps standards, ensuring performance, scalability, and maintainability.