VIETNAM GENERAL CONFEDERATION OF LABOUR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**PHAM HOANG SON – 521H0476**

**NGUYEN DINH VIET HOANG - 522H0120**

**MIDTERM REPORT**

**WEB APPLICATION DEVELOPMENT USING NODEJS**

**HO CHI MINH CITY, YEAR 2024**

VIETNAM GENERAL CONFEDERATION OF LABOUR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**PHAM HOANG SON – 521H0476**

**NGUYEN DINH VIET HOANG - 522H0120**

**MIDTERM REPORT**

**WEB APPLICATION DEVELOPMENT USING NODEJS**

Advised by

**MsC. MAI VAN MANH**

**HO CHI MINH CITY, YEAR 2024**

**ACKNOWLEDGEMENT**

We sincerely thank MsC. Mai Van Manh for teaching us the Web Application Development Using NodeJS course with great enthusiasm. We want to express our deep appreciation for the dedication and professional knowledge that you shared with us. Through your classes, we gained a better understanding of the fundamental aspects of the Web Application Development Using NodeJS, thanks to your detailed explanations and practical applications. You helped us grasp the knowledge and apply it effectively. Finally, we extend our heartfelt gratitude to MsC. Mai Van Manh for your commitment and invaluable support throughout our learning journey in this course. The skills and knowledge we acquired will continue to impact our future development. We sincerely thank you and wish your health, success, and happiness.

*Ho Chi Minh City, October 21, 2024*

*Authors:*

*Pham Hoang Son*

*Nguyen Dinh Viet Hoang*

**DECLARATION OF AUTHORSHIP**

We hereby declare that this thesis was carried out by ourselves under the guidance and supervision of MsC. Mai Van Manh; and that the work and the results contained in it are original and have not been submitted anywhere for any previous purposes. The data and figures presented in this thesis are for analysis, comments, and evaluations from various resources by our own work and have been duly acknowledged in the reference part.

In addition, other comments, reviews and data used by other authors, and organizations have been acknowledged, and explicitly cited.

We will take full responsibility for any fraud detected in my thesis. Ton Duc Thang University is unrelated to any copyright infringement caused on my work (if any).

*Ho Chi Minh City, May 12, 2024*

*Authors:*

*Pham Hoang Son*

*Nguyen Dinh Viet Hoang*

**ABSTRACT**

**TABLE OF CONTENTS**

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS

# CHAPTER 1: INTRODUCTION TO DOCKER

## *Define what Docker is and explain why it is important in modern web development:*

Docker is an open-source platform that allows you to build, deploy, and manage programs within containers. Containers are standalone entities that include everything needed to run an application, including source code, libraries, and the operating system environment. Docker isolates applications from the underlying infrastructure, making deployment and scaling simple and consistent independent of environment (development, testing, or production). Docker is essential for modern web development as it enables consistency and reproducibility across all systems, including personal development computers and production settings. This solves the "it works on my machine but not on yours" problem, which is a significant challenge in software development. Containers require fewer system resources than virtual machines, making it easier to scale applications as demand grows. Docker also offers efficient resource management, which lowers hardware costs and improves performance. Also, Docker works with CI/CD (Continuous Integration/Continuous Deployment) pipelines to automate and speed up deployment by packaging applications and dependencies. Docker allows for seamless application portability across several environments, including personal computers, on-premise servers, and the cloud.

## *Key Concepts:*

a. Containers:

Containers are “packages” that contain an application and all the necessary components (libraries, dependencies) needed for it to run. In addition, containers provide an isolated environment but share the kernel with the host operating system, optimizing resource usage. Advantages of containers is they are lightweight, fast, and highly reusable. Compared to virtual machines, containers start faster and use fewer resources because they don't require a full operating system installation.

b. Images:

Docker images are immutable files that contain the source code and the environment needed to run containers. An image is a snapshot of an application at a specific point in time and is used to create containers. Besides, each image can consist of multiple layers. These layers are based on changes made during the image creation process (via a Dockerfile). Furthermore, images can be stored and shared through registries such as Docker Hub.

c. Dockerfile:

A Dockerfile is a text file containing instructions that Docker uses to build an image. The Dockerfile automates the process of creating images and defines the exact environment needed for an application.

* **Example of a Dockerfile:**

Dockerfile:

FROM node:14

WORKDIR /app

COPY . .

RUN npm install

CMD ["npm", "start"]

In this example, the Dockerfile sets up a Node.js environment, copies the application source code, installs dependencies, and starts the application using “npm start”.

d. Docker Compose:

Docker Compose is a tool that helps manage multiple containers in a complex application which allows you to define containers and how they interact with each other in a configuration file called docker-compose.yml. With Docker Compose, you can start, stop, or manage all the containers of an application with a single command.

* **Example of Docker Compose**:

YAML:

version: '3'

services:

web:

image: nginx

ports:

- "80:80"

database:

image: postgres

environment:

POSTGRES\_PASSWORD: example

Here, Docker Compose defines two services for the application: one container running the NGINX web server and another container running a PostgreSQL database.

e. Container Orchestration:

Container orchestration is the process of automating the deployment, management, scaling, and coordination of containers in a distributed system. The most popular tool for this is Kubernetes. Moreover, Kubernetes manages containers across a cluster of servers, ensuring that the application runs smoothly, automatically distributes resources, load balances, and restarts failed containers. It makes it easy to scale applications as needed and ensures high availability.

# CHAPTER 2: THEORETICAL SURVEY:

# REFERENCES

1. [What is Docker? | Docker Docs](https://docs.docker.com/get-started/docker-overview/)
2. [Docker for Web Developers: Starting with the Basics | Docker](https://www.docker.com/blog/docker-for-web-developers/)
3. [What is a Container? | Docker](https://www.docker.com/resources/what-container/)
4. [Dockerfile reference | Docker Docs](https://docs.docker.com/reference/dockerfile/)
5. [Docker Compose | Docker Docs](https://docs.docker.com/compose/)
6. [What is container orchestration?](https://www.redhat.com/en/topics/containers/what-is-container-orchestration)