



# **AMD201**

## **Advanced Microservices Development & Deployment**

**Student's Name:** Vu Thanh Nguyen

**Student's ID:** GCH221117 - 001343200

**Submission Date:** 28/04/2025

**Group Number:** 4

## Table of Contents

<b>I. Introduction</b>	3
<b>1.1 Group Introduction</b>	3
<b>1.2 Project Introduction</b>	3
<b>II. Evaluation</b>	3
<b>2.1 Personal Contribution</b>	3
<b>2.2 Challenges &amp; Solutions</b>	3
<b>2.3 Learning Outcomes</b>	4
<b>2.4 Member Evaluation</b>	4
<b>2.5 Assignment Evaluation</b>	4
<b>Appendices Github Link:</b>	5

# I. Introduction

## 1.1 Group Introduction

Our team **KCN ( NguyenAnhKhoi, HoangVanCuong, VuThanhNguyen)** , is a dedicated group of 3 members with diverse technical skills and a shared commitment to delivering an innovative project. Each member plays a key role in achieving project goals through effective collaboration and problem-solving.

## 1.2 Project Introduction

The project, KCN-sortenlink, aims to develop an efficient and scalable system that converts long URLs into shorter, user-friendly links. It addresses challenges related to sharing long URLs across platforms with character limits and seeks to improve overall user experience.

We leveraged **Express.js** for backend API development, **MongoDB** for data persistence, and **React.js** for the frontend web application.

Our solution focuses on performance, usability, and scalability, aiming to provide a reliable and convenient tool for individuals and organizations to manage their URLs effectively.

# II. Evaluation

## 2.1 Personal Contribution

- **Role:** Frontend Developer
- **Key Contributions:**
  - System Architecture Design – Designed the overall backend structure using Express.js and MongoDB, ensuring scalability and performance.
  - API Development – Developed RESTful APIs for URL shortening, redirection, and link management, with input validation and error handling.
  - Database Integration – Set up MongoDB collections and schemas to efficiently store and retrieve URL data, ensuring fast lookups.

## 2.2 Challenges & Solutions

- **Challenge 1:** Generating unique short codes to prevent URL conflicts.
  - Solution: Implemented a random string generation algorithm with collision checks in MongoDB to ensure code uniqueness.
- **Challenge 2:** Handling high-traffic scenarios and optimizing performance.
  - Solution: Designed the system to be horizontally scalable and optimized MongoDB queries; planned to integrate Redis caching for frequently accessed URLs in future iterations.

## 2.3 Learning Outcomes

- **Technical Skills:**
  - Express.js (Backend Development)
  - MongoDB (NoSQL Database Management)
  - RESTful API Design
  - React.js (Frontend Basics)
  - Mongoose ODM (Schema Management)
  - API Testing (Postman, Jest)
- **Soft Skills:**
  - Effective Team Collaboration
  - Critical Problem-Solving
  - Project Planning and Time Management
  - Adaptability and Continuous Learning

## 2.4 Member Evaluation

- **Teamwork:**

Team KCN demonstrated excellent collaboration throughout the project. Members frequently exchanged ideas, supported each other in solving problems, and divided tasks clearly and effectively. Strong teamwork spirit helped maintain steady project progress.
- **Contribution:**

Each member made significant contributions, including backend system design, API development, frontend interface building, and system testing. Proactive participation and a strong sense of responsibility from every member ensured that the project met both quality and requirement standards.
- **Attitude:**

All members showed a positive, enthusiastic, and responsible attitude. They maintained a strong willingness to learn, adapt to new technologies, and overcome technical challenges throughout the project.

## 2.5 Assignment Evaluation

- **Suitability:**

The assignment topic — building a URL shortener system — was highly suitable for practicing real-world application development skills. It required a combination of backend, frontend, and database technologies, helping students reinforce their system design knowledge and full-stack development skills.
- **Difficulty:**

The level of difficulty was appropriate — challenging enough to promote learning, yet manageable within the given timeframe. Requirements like API design, data storage handling, and web interface development allowed the team to strengthen both their programming abilities and system thinking.

- **Suggestions:**

To further enhance realism, the assignment could include requirements for deploying the service on cloud platforms (e.g., AWS, Azure) or integrating caching technologies like Redis. Additionally, requiring a performance analysis under high-load conditions would better prepare students for real-world projects.

## **Appendices Github Link:**

<https://github.com/hoangcuong25/coursework-AMD>