PHAM VAN HAU

0375684441 | phamvanhau692002vip@gmail.com | https://github.com/longsoisuaxe1a

EDUCATION

I am currently a 4th-year student majoring in Software Engineering at the Industrial University of Ho Chi Minh City (IUH).

COURSE PROJECTS

Project: Management_BookStore_Microservice

Description:

This project is a bookstore management system developed based on the microservices architecture. The main goal is to decompose the functional components of the application into small independent services, enhancing flexibility and efficient management. The project focuses on deploying and managing microservices through the integration of technologies such as GitLab, Jenkins, and Docker.

Technologies:

- GitLab
- Jenkins
- Docker

Features:

1. Building and deploying microservices to manage various aspects of the bookstore, such as book management, order management, user management, etc.
2. Utilizing Jenkins for automating testing and deployment processes.
3. Leveraging Docker for containerizing and running applications.

Results:

1. Increased flexibility and efficient management through the adoption of the microservices architecture.
2. Automated testing and deployment processes improve development and deployment workflows.

Experience:

- Experience in deploying and managing microservices.
- Proficiency in GitLab, Jenkins, and Docker for establishing effective development and deployment pipelines.

Source Code: https://github.com/longsoisuaxe1a/Management_BookStore_Microservice

---

Project: Connect Node.js AWS CRUD

Description:

This project involves the development of a CRUD (Create, Read, Update, Delete) application using Node.js and AWS (Amazon Web Services). It aims to establish connectivity between a Node.js application and various AWS services, enabling CRUD operations on data stored within AWS.

Technologies:

Node.js, AWS (Amazon Web Services)

Features:

- Connectivity with AWS services such as DynamoDB, S3, EC2, etc.
- Implementation of CRUD operations on data stored within AWS.
- Likely involves authentication and authorization mechanisms for secure access to AWS resources.

Results:

- Improved scalability and flexibility by leveraging AWS services.
- Potential for enhanced data management and security through AWS features.

Experience:

- Experience in full-stack development with Node.js and AWS integration.
- Proficiency in handling asynchronous operations in Node.js.
- Familiarity with AWS SDKs and APIs for service integration.

Source Code: https://github.com/longsoisuaxe1a/connect-nodejs-aws-crud

---

Project: Node.js MongoDB API

Description:

This project involves the development of an API using Node.js and MongoDB. The main objective is to create an API to perform CRUD (Create, Read, Update, Delete) operations on MongoDB database.

Technologies:

Node.js, MongoDB

Features:

- Handling HTTP requests to perform CRUD operations on MongoDB.
- Interacting with MongoDB database to store and retrieve data.
- Providing API endpoints to allow other applications to communicate and work with data.
- Results:
- Providing a flexible and efficient approach to manage data within the application.
- Potential for easy scalability and updates of the project in the future.
- Experience:
- Experience in developing APIs using Node.js and MongoDB.
- Proficiency in handling HTTP requests and interacting with MongoDB database.
- Understanding of basic concepts of RESTful API and API design.
- **Source Code:** https://github.com/longsoisuaxe1a/nodejs-mongodb-api

---

Project: Node.js Gen-Chat

Description:

The "nodejs-genchat" part of the Gen-Chat project on GitHub serves as the backend of the chat application. Its primary task is to build server-side logic and APIs to implement the features of the chat system.

Technologies:

- Node.js: Utilized for server-side scripting, handling client requests, and executing logical operations.
- Express.js: Employed to create APIs and handle HTTP requests from the client side.
- Socket.IO: Utilized to provide real-time communication between clients and the server, enabling instant messaging.

Features:

- Real-time communication handling: Utilizes Socket.IO to establish and manage real-time connections between clients and the server.

- User management: Implements authentication and user management features, including login, registration, and authentication.
- Message handling: Executes logic to send, receive, and store messages from users in the database.

Experience:

- Backend development with Node.js: Working on "nodejs-genchat" provides practical experience in developing backend solutions using Node.js, including API construction and client request handling.
- Real-time communication handling: Understanding how to create and manage real-time connections between clients and the server is enhanced through developing this chat application.
- User and message management: Developing features for user and message management enhances experience in building basic functionalities of a chat application.
- **Source Code:** https://github.com/longsoisuaxe1a/Gen-Chat

---

TECHNICAL SKILLS

- Programming: HTML, CSS, Java, JavaScript, Node.js, React.js
- Frameworks: Bootstrap, jQuery, Spring Boot
- Databases: SQL Server, MySQL, MongoDB
- AWS Web Services: EC2, S3, DynamoDB

SOFT SKILLS

- Communication
- Teamwork
- Problem-solving
- Adaptability to the working environment
- Creative thinking, quick learning, and flexibility
