HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN 1

BÁO CÁO LẦN I

MÔN HỌC: THỰC TẬP TỐT NGHIỆP

Doanh nghiệp: Công ty cổ phần VCCORP

Giảng viên hướng dẫn: Đỗ Thị Liên

Nhóm: 25

Họ tên: Hoàng Văn An

Mã sinh viên: B20DCCN045

Số điện thoại: 0867865001

Email: vananhoang10052002@gmail.com

Hà Nội – 2024

NHẬN XÉT CỦA ĐƠN VỊ THỰC TẬP

……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……

NHẬN XÉT CỦA GIẢNG VIÊN

………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………
………

MỤC LỤC

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật nghữ	Ý nghĩa
API	Application Programming Interface : Giao diện lập trình ứng dụng
Aerospike	Aerospike là một hệ thống cơ sở dữ liệu NoSQL hiệu năng cao được thiết kế để xử lý khối lượng dữ liệu lớn và cung cấp tốc độ truy xuất nhanh, hỗ trợ lưu trữ vĩnh viễn và khả năng mở rộng cao
Hbase	HBase là một cơ sở dữ liệu phân tán, mã nguồn mở, được thiết kế để cung cấp khả năng lưu trữ và truy xuất dữ liệu có cấu trúc lớn với hiệu suất cao, chạy trên hệ thống file HDFS của Hadoop.
Kafka	Kafka là một nền tảng streaming phân tán mạnh mẽ, được thiết kế để xử lý và truyền tải lượng lớn dữ liệu theo thời gian thực với độ tin cậy cao và khả năng mở rộng linh hoạt.
Spark	Spark là một framework xử lý dữ liệu phân tán mạnh mẽ, hỗ trợ xử lý dữ liệu lớn với hiệu suất cao thông qua các API dễ sử dụng cho SQL, streaming, machine learning, và graph processing.
Docker	Docker là một nền tảng mã nguồn mở giúp tự động hóa việc triển khai ứng dụng bên trong các container phần mềm, đảm bảo tính nhất quán, di động và hiệu quả trong việc phát triển và vận hành phần mềm.
MongoDB	MongoDB là một cơ sở dữ liệu NoSQL dạng tài liệu, cung cấp khả năng lưu trữ và truy vấn dữ liệu linh hoạt và mở rộng dễ dàng.
Design pattern	Design pattern là các giải pháp thiết kế tái sử dụng để giải quyết các vấn đề phổ biến trong phát triển phần mềm.
Hadoop	Hadoop là một khung phần mềm mã nguồn mở cho phép xử lý và lưu trữ khối lượng lớn dữ liệu phân tán trên các cụm máy tính sử dụng mô hình lập trình đơn giản.
Zookeeper	Zookeeper là một dịch vụ đồng bộ hóa phân tán cung cấp các công cụ để quản lý cấu hình, phối hợp dịch vụ và giữ cho các ứng dụng phân tán đồng bộ.
Kraft	Kraft là một thư viện mã nguồn mở giúp triển khai và quản lý các dịch vụ kết nối với Kafka trong môi trường Kubernetes.
RabbitMQ	RabbitMQ là một hệ thống quản lý hàng đợi tin nhắn mã nguồn mở, hỗ trợ giao tiếp giữa các ứng dụng và dịch vụ thông qua các hàng đợi tin nhắn.

PHẦN I: GIỚI THIỆU CHUNG ĐƠN VỊ THỰC TẬP

Chương này cung cấp một cái nhìn tổng quan về đơn vị thực tập, bao gồm lịch sử hình thành và phát triển, tầm nhìn và sứ mệnh của công ty. Chương này cũng sẽ giới thiệu các sản phẩm và dịch vụ chính mà công ty cung cấp, nhấn mạnh vào những giá trị cốt lõi và cam kết của công ty trong việc mang lại các giải pháp công nghệ tiên tiến và hiệu quả cho khách hàng.

1.1. Thông tin về đơn vị thực tập:

1.1.1. Giới thiệu về VCCorp:

- Được thành lập vào năm 2006, sau 18 năm phát triển, Công ty CP VCCorp (VCCorp) là một trong những công ty tiên phong trong lĩnh vực công nghệ và công nghệ cao ở Việt Nam. VCCorp đã xây dựng và ứng dụng thành công hạ tầng cloud computing có khả năng lưu trữ và tính toán lượng dữ liệu lớn (Big Data, Data mining) hỗ trợ cho nền tảng công nghệ quảng cáo trực tuyến Admicro, phục vụ trên 50 triệu người dùng (chiếm trên 90% tổng số người dùng Internet tại Việt Nam) với hơn 200 website uy tín, gần 30 website trong số đó thuộc sở hữu độc quyền, phục vụ hàng tỷ lượt xem mỗi tháng. Ngoài ra, VCCorp nằm trong top 3 công ty phát hành game trên nền tảng di động với thương hiệu Sohagame, và mới đây hệ sinh thái chuyển đổi số Bizfly đang được VCCorp phát triển và triển khai mạnh mẽ. Đồng thời VCCorp cũng là đơn vị xây dựng Lotus - mạng xã hội do người Việt Nam làm chủ.

Hình 1.1. Tổng quan về các sản phẩm của VCCorp

1.2. Cơ sở thực thập:

1.2.1. Thông tin thực tập

- Tên doanh nghiệp: Công ty cổ phần VCCORP
- Vị trí thực tập: Java Backend Developer
- Khối được phân: Khối Adtech của VCCORP
- Leader, mentor hướng dẫn: Leader Ngô Văn Vĩ
- Hình thức thực tập: Tại văn phòng công ty
- Mô tả công việc của team:
  - Xử lí các phần backend Server của các hệ thống phân tán, các hệ thống dữ liệu, data minning.

- Tối ưu, xử lí dữ liệu.

1.2.2. Đội ngũ hướng dẫn thực tập sinh:

- Leader Ngô Văn Vĩ:
  - Chuyên gia với kiến thức chuyên sâu về các công nghệ mới nhất và xu hướng phát triển trong ngành.

1.2.3. Cơ sở vật chất:

- VCCorp có văn phòng được thiết kế theo tiêu chuẩn công nghệ cao, với không gian mở và các khu vực làm việc riêng biệt. Thiết kế văn phòng tối ưu hóa sự linh hoạt và sự giao tiếp giữa các bộ phận..
- Công nghệ, thiết bị, cập nhật liên tục, đảm bảo nhân viên luôn tiếp cận với những công nghệ mới nhất.
- Cơ sở vật chất của VCCorp được đảm bảo về mặt an ninh và an toàn, với hệ thống bảo vệ chuyên nghiệp và các biện pháp phòng cháy chữa cháy đầy đủ.

1.2.4. Dịch vụ hỗ trợ thực tập sinh:

- Đào tạo và hướng dẫn:
  - Cung cấp chương trình đào tạo ban đầu để giúp thực tập sinh làm quen với công việc và các công cụ cần thiết.
  - Được hướng dẫn bởi các chuyên gia có kinh nghiệm trong ngành, giúp thực tập sinh nâng cao kỹ năng chuyên môn.
- Môi trường làm việc chuyên nghiệp:
  - Cung cấp môi trường làm việc hiện đại, tiện nghi với đầy đủ trang thiết bị cần thiết.
  - Không gian làm việc thân thiện, khuyến khích sự sáng tạo và hợp tác giữa các nhân viên.
- Phản hồi và đánh giá:
  - Thường xuyên cung cấp phản hồi và đánh giá hiệu quả làm việc, giúp thực tập sinh nhận biết được điểm mạnh và điểm cần cải thiện.
  - Hỗ trợ phát triển cá nhân thông qua các buổi mentoring và coaching.
- Cơ hội phát triển nghề nghiệp:
  - Tạo điều kiện để thực tập sinh tham gia vào các dự án thực tế, tích lũy kinh nghiệm quý báu.
  - Cơ hội trở thành nhân viên chính thức của VCCorp sau khi hoàn thành chương trình thực tập xuất sắc.

1.2.5. Thời gian thực tập:

- Thực tập từ 26/6/2024 đến 11/8/2024
- Lên công ty theo lịch đã dăng kí, buổi sáng từ 9h đến 12h, buổi chiều từ 13h30 đến 18h

PHẦN II: **NỘI DUNG CÔNG VIỆC TRONG KÌ THỰC TẬP:**

Nội dung thực tập trong 7 tuần (trao đổi với leader):

- Tuần 1 (24/6 – 30/6): Tìm hiểu, làm bài tập và giải đáp thắc mắc về java basic, java core.
- Tuần 2 (30/6 – 6/7): Tìm hiểu, làm bài tập và giải đáp thắc mắc về các công nghệ được áp dụng trong dự án thực tế của công ty.

- Tuần 3 - 4(6/7 – 12/7): Tìm hiểu, làm bài tập và giải đáp thắc mắc về các công nghệ được áp dụng trong dự án thực tế của công ty.
- Tuần 5 – Tuần 7(12/7 – 11/8): Leader sẽ giao một dự án nhỏ cho mỗi thành viên trong nhóm thực tập, với yêu cầu chi tiết và tư vấn hướng dẫn. Nếu đạt yêu cầu và đánh giá của leader thi sẽ được giao vào một dự án thực tế.

2.1. NỘI DUNG CÔNG VIỆC TUẦN 1 (24/6 – 1/7):

2.1.1. Tìm hiểu, làm bài tập và giải đáp các thắc mắc về java basic:

- Task 1: Tìm hiểu và viết báo cáo về các thuật toán tìm kiếm và sắp xếp, độ phức tạp thuật toán, nghiên cứu làm bài tập với độ phức tạp thuật toán nhỏ nhất.
- Task 2: Tìm hiểu và viết báo cáo về http và url trong phương thức HTTP, Tìm hiểu về xử lí chuỗi trong java, cách sử dụng regex đê cho phép bạn tìm kiếm, so khớp, và thao tác với các mẫu chuỗi văn bản cụ thể, nghiên cứn làm bài tập về validate url hợp lệ trong HTTP.
- Task 3: Tìm hiểu và viết báo cáo về Prefix Sum và ứng dụng trong bài toán counting sort
- Task 4: Tìm hiểu và viết báo cáo về bài toán xếp các từ thành câu có nghĩa và phương pháp đệ quy kết hợp với lưu trữ kết quả trung gian (memoization) để tránh tính toán lại các phần đã được xử lý, áp dụng vào bài toán viết một phương thức add các khoảng trắng vào chuỗi s sao cho thành các câu có thể, với một từ điển các từ đã được cho sẵn
- Task 5: Tìm hiểu và viết báo cáo về các design pattern trong java. Mô tả và implement lại chúng (phân tích so sánh dựa trên 1 case bài toán thực tế)
- Task 6: Tìm hiểu về các quy tắc clean code và viết báo cáo

2.1.2. Kết quả đạt được:

- Hiểu và cài đặt các thuật toán
- Thực hiện các bài tập với độ phức tạp thuật toán nhỏ nhất.
- Hiểu rõ cơ bản về HTTP, các phương thức HTTP (GET, POST, PUT, DELETE), tìm hiểu về cấu trúc URL và cách sử dụng.
- Thực hiện bài tập validate URL hợp lệ trong HTTP.
- Hiểu rõ thuật toán Prefix Sum và cách áp dụng nó, cài đặt và thử nghiệm thuật toán Counting Sort với ứng dụng của Prefix Sum.
- Hiểu về phương pháp đệ quy kết hợp với lưu trữ kết quả trung gian (memoization).
- Tìm hiểu và mô tả các design pattern phổ biến như Singleton, Factory, Adapter, Chain of Responsibility, cài đặt và thử nghiệm các design pattern này dựa trên các case thực tế.
- Hiểu rõ các nguyên tắc cơ bản của Clean Code, thực hiện các bài tập để viết mã sạch và dễ bảo trì.

2.1.3. Bài học kinh nghiệm:

- Hiểu rõ hơn về cách đánh giá độ phức tạp của các thuật toán và áp dụng vào các bài toán thực tế.
- Hiểu rõ hơn về đệ quy và memoization, và cách áp dụng vào các bài toán xử lý chuỗi.
- Nắm bắt được các design pattern và cách áp dụng chúng vào các dự án thực tế.
- Nhận ra tầm quan trọng của Clean Code và cách áp dụng nó để tạo ra mã nguồn dễ hiểu và dễ bảo trì.

2.2. NỘI DUNG CÔNG VIỆC TUẦN 2 (1/7 – 7/7)

2.2.1. Tìm hiểu về các công nghệ thực tế được sử dụng trong công ty

- **Track 1: Tìm hiểu và viết báo cáo về Aerospike, sau đó thực hiện các task sử dụng công nghệ đã tìm hiểu đó, mentor giải đáp các thắc mắc về công nghệ**
    - Cài đặt Aerospike
    - Ghi 1 bản ghi vào Aerospike (Ghi vĩnh viễn/có expire time)
    - Đọc 1 bản ghi từ Aerospike
    - Đọc nhiều bản ghi từ Aerospike
    - Sử dụng AQL
    - Tìm hiểu về các operation khác của Aerospike
    - Tìm hiểu về EventLoop và Callback trong Aerospike
    - Scan và Query
    - Tìm hiểu index trong Aerospike
    - So sánh giữa Aerospike với Redis
    - Tìm hiểu các chiến lược caching cho cả đọc và ghi

- **Track 2: Tìm hiểu và viết báo cáo về HBase, sau đó thực hiện các task sử dụng công nghệ đã tìm hiểu đó, mentor giải đáp các thắc mắc về công nghệ**
    - Cài đặt HBase (có thể dùng docker)
    - Thao tác bằng HBase shell
    - Thao tác bằng Java API
    - Get
    - Exist bin
    - Exist row
    - Put
    - Delete bin
    - Delete Row
    - Bulk
    - Time To Live
    - Scan + Filter

- Pagination using HBase Scan
- Tìm hiểu vai trò của ZooKeeper trong HBase
- Tìm hiểu về Compact trong Hbase
- Tìm hiểu các thành phần cấu hình khi cài đặt Hbase

2.2.2. Những kết quả đạt được:

- Hoàn thành cài đặt Aerospike và HBase.
- Thực hiện thành công các thao tác cơ bản với Aerospike và HBase.
- Viết báo cáo chi tiết về các công nghệ, cách cài đặt, và các thao tác cơ bản.
- So sánh Aerospike và Redis, nắm rõ các ưu nhược điểm của từng công nghệ.
- Áp dụng các chiến lược caching hiệu quả.

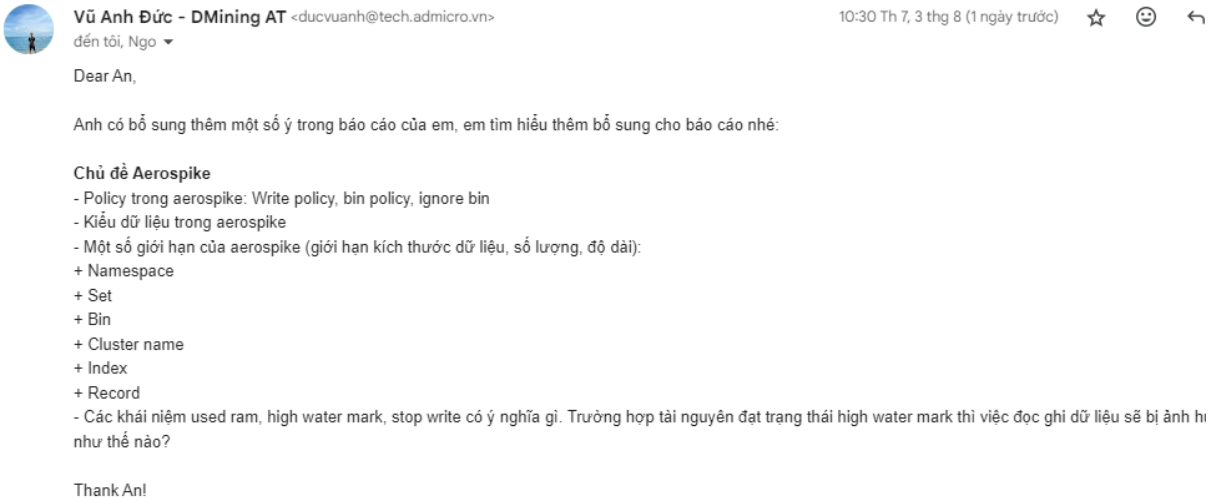
2.2.3. Những điều chưa đạt được:

- Chưa tối ưu hóa toàn diện cho các thao tác với dữ liệu lớn trong Aerospike và HBase.
- Chưa triển khai các kịch bản phức tạp sử dụng EventLoop và Callback trong Aerospike.
- Cần tìm hiểu thêm về các mô hình dữ liệu phức tạp và các phương pháp tối ưu hóa hiệu suất trong HBase.

2.2.4. Bài học kinh nghiệm:

- Cần tìm hiểu kỹ lưỡng về cấu trúc và mô hình dữ liệu của từng công nghệ trước khi triển khai.
- Việc sử dụng Docker giúp giảm thiểu thời gian cài đặt và cấu hình môi trường phát triển.
- Cần thường xuyên tham khảo tài liệu và hướng dẫn chính thức của từng công nghệ để nắm bắt các tính năng mới và các phương pháp tối ưu hóa.
- Việc so sánh và lựa chọn công nghệ cần dựa trên yêu cầu cụ thể của dự án và khả năng mở rộng trong tương lai.

2.2.5. Review:



Vũ Anh Đức - DMining AT <ducvuanh@tech.admicro.vn> 10:30 Th 7, 3 thg 8 (1 ngày trước)
đến tôi, Ngo

Dear An,

Anh có bổ sung thêm một số ý trong báo cáo của em, em tìm hiểu thêm bổ sung cho báo cáo nhé:

Chủ đề Aerospike
- Policy trong aerospike: Write policy, bin policy, ignore bin
- Kiểu dữ liệu trong aerospike
- Một số giới hạn của aerospike (giới hạn kích thước dữ liệu, số lượng, độ dài):
+ Namespace
+ Set
+ Bin
+ Cluster name
+ Index
+ Record
- Các khái niệm used ram, high water mark, stop write có ý nghĩa gì. Trường hợp tài nguyên đạt trạng thái high water mark thì việc đọc ghi dữ liệu sẽ bị ảnh hư như thế nào?

Thank An!

2.3. NỘI DUNG CÔNG VIỆC TUẦN 3 - 4 (7/7 – 21/7):

2.3.1. Tìm hiểu về các công nghệ thực tế được sử dụng trong công ty

- **Track 3: Tìm hiểu và viết báo cáo về Kafka, sau đó thực hiện các task sử dụng công nghệ đã tìm hiểu đó, mentor giải đáp các thắc mắc về công nghệ.**
    - Cài đặt Kafka
    - Produce message
    - Consume message
    - Consumer group
    - Tìm hiểu về partition và replicate trong Kafka
    - Tìm hiểu vai trò của ZooKeeper trong Kafka (tìm hiểu cả các loại khác thay thế zookeeper)
    - So sánh với rabitmq

- **Track 4: Tìm hiểu và viết báo cáo về Spark, sau đó thực hiện các task sử dụng công nghệ đã tìm hiểu đó, mentor giải đáp các thắc mắc về công nghệ**
    - Tìm hiểu cách thức hoạt động của Spark
    - Tìm hiểu RDD
    - Tìm hiểu mô hình MapReduce
    - Tìm hiểu Spark Streaming
    - Tìm hiểu HDFS, Parquet file
    - Spark SQL
    - Spark SQL Optimization

2.3.2. Kết quả đạt được:

- Hoàn thành cài đặt và cấu hình Kafka và Spark.
- Thực hiện thành công các thao tác cơ bản với Kafka và Spark.
- Viết báo cáo chi tiết về các công nghệ, cách cài đặt, và các thao tác cơ bản.
- So sánh Kafka với RabbitMQ, nắm rõ các ưu nhược điểm của từng công nghệ.
- Hiểu cách tối ưu hóa truy vấn và xử lý dữ liệu với Spark.

2.3.3. Những điều chưa đạt được

- Chưa tối ưu hóa toàn diện cho các thao tác với dữ liệu lớn trong Kafka và Spark.
- Chưa triển khai các kịch bản phức tạp sử dụng Spark Streaming và các mô hình phức tạp trong Kafka.
- Cần tìm hiểu thêm về các phương pháp tối ưu hóa hiệu suất trong Kafka và Spark.

2.2.3. Bài học kinh nghiệm

- Cần tìm hiểu kỹ lưỡng về cấu trúc và mô hình dữ liệu của từng công nghệ trước khi triển khai.
- Việc sử dụng Docker giúp giảm thiểu thời gian cài đặt và cấu hình môi trường phát triển.

- Cần thường xuyên tham khảo tài liệu và hướng dẫn chính thức của từng công nghệ để nắm bắt các tính năng mới và các phương pháp tối ưu hóa.
- Việc so sánh và lựa chọn công nghệ cần dựa trên yêu cầu cụ thể của dự án và khả năng mở rộng trong tương lai.

2.4. NỘI DUNG CÔNG VIỆC TUẦN 5 (21/7 – 28/7):

2.4.1. Tìm hiểu về công nghệ và project:

- **Track 5: Tìm hiểu về MongoDB và viết báo cáo làm các task liên quan:**
    - Replica
    - Sharding
    - WiredTiger (storage)
    - xem phần pluggable storage
    - Deployment and administration
    - Design patterns
    - Queue, tree, transactions
    - Optimization query
    - Tìm hiểu về hệ thống Elastic Search
    - Tìm hiểu Debezium
- Công nghệ trên thường được sử dụng trong các hệ thống 1uản lý dữ liệu, hệ thống phân phối quảng cáo, hệ thống phân tích và báo cáo quảng cáo của công ty. Những công nghệ này giúp tăng cường hiệu suất, độ tin cậy, và khả năng mở rộng của các hệ thống AdTech, giúp công ty quản lý và tối ưu hóa các chiến dịch quảng cáo hiệu quả hơn.

2.4.2. Kết quả đã đạt được:

- Hiểu và triển khai mô hình Replica Set, kiểm tra và xác nhận việc failover giữa các node trong Replica Set, thực hiện các thao tác đọc/ghi trong môi trường Replica Set.
- Tìm hiểu cách phân mảnh dữ liệu (sharding) để quản lý khối lượng dữ liệu lớn.
- Hiểu về WiredTiger Storage Engine và các ưu điểm của nó, cấu hình và kiểm tra hiệu suất của WiredTiger so với các storage engine khác.
- Nghiên cứu về khả năng thay đổi storage engine trong MongoDB, thử nghiệm với các storage engine khác nhau và so sánh hiệu suất.
- Triển khai MongoDB trong các môi trường khác nhau (local, cloud).
- Áp dụng các mẫu thiết kế Designe pattern phổ biến trong việc xây dựng ứng dụng với MongoDB.
- Cài đặt và kiểm tra các cấu trúc hàng đợi (queue) và cây (tree) trong MongoDB, sử dụng các giao dịch (transactions) để đảm bảo tính nhất quán dữ liệu.
- Tối ưu hóa các truy vấn MongoDB để cải thiện hiệu suất, tìm hiểu và tích hợp Elastic Search với MongoDB.
- Nghiên cứu Debezium và cách nó hoạt động với MongoDB, cấu hình và thử nghiệm việc stream dữ liệu từ MongoDB sang các hệ thống khác.

2.4.3. Kết quả chưa đạt được:

- Vẫn còn khó khăn trong việc tối ưu hóa failover để giảm thiểu downtime.
- Chưa thể kiểm tra hết các kịch bản sử dụng WiredTiger trong môi trường sản xuất.
- Chưa có đủ thời gian để thử nghiệm sâu hơn với nhiều loại storage engine khác nhau.
- Còn gặp khó khăn trong việc đồng bộ dữ liệu real-time.
- Tích hợp Elastic Search với MongoDB vẫn còn một số hạn chế về hiệu suất.

2.4.4. Bài học kinh nghiệm:

- Tìm hiểu kỹ về các công cụ và công nghệ mới để đảm bảo hiệu suất và tính ổn định.
- Luôn cập nhật và áp dụng các best practices trong quản trị và triển khai hệ thống MongoDB.
- Luôn kiểm tra và tối ưu hóa cấu trúc dữ liệu trước khi triển khai hệ thống.

2.5. NỘI DUNG CÔNG VIỆC TUẦN 6 – 7 (28/7 – 11/8):

2.5.1. Project:

- **Xây dựng một hệ thống search engine:**
    - Có luồng stream CDC (Data Change Capture) từ một hệ thống database khác
    - Có tính năng autocomplete (tìm hiểu thuật toán liên quan)293989
    - Có tính năng search, và có thể tùy chọn ranking
    - Có giao diện thực hiện việc search

- Tình trạng: Đang thực hiện công việc, bước đầu xây dựng project, thiết kế database.