Project Documentation

Introduction

The **Rainbow Birthday Discount** project is designed to send discounts to customers who have their upcoming birthday within one week. The system is built using NestJS, a powerful Node.js framework, and utilizes various services to accomplish its functionalities. This documentation provides an analytical approach, observations, and details the architecture and infrastructure of the project.

Analytical Approach

The analytical approach for the **Rainbow Birthday Discount** project is centered around providing a seamless and personalized experience to customers. The main goals of the project include:

1. **User Management:** Allow users to sign up and log in to their accounts securely.
2. **JWT Authentication:** Ensure secure authentication and authorization using JSON Web Tokens (JWT) to protect APIs and restrict access to certain endpoints.
3. **Discount Generation:** Generate unique discount codes for eligible users with upcoming birthdays.
4. **Recommender System:** Recommend products to customers based on their purchase and view history.
5. **Email Notification:** Send personalized birthday discount emails to eligible users.
6. **Scalability:** Design an architecture that can handle high traffic and efficiently process the message queue.

Observations

1. **Database and ORM:** The project utilizes PostgreSQL as the database and Prisma as the Object-Relational Mapping (ORM) tool to interact with the database efficiently.
2. **Message Queue:** The project uses Bull as the message queue for processing emails asynchronously. Redis serves as the broker for the message queue.
3. **Service Segmentation:** The project is divided into multiple services to separate concerns and improve maintainability. Each service is responsible for specific functionalities.
4. **Micro-**services Architecture: The project is designed in a way that, each section can be scalable and have multiple instances of different services. All of these different services communicate with each other through apis (further implementation can be done using RPC).

Architecture and Infrastructure

The **Rainbow Birthday Discount** project follows a microservices architecture to ensure modularity, scalability, and maintainability. The architecture consists of the following services:

1. **rainbow-birthday-discount:** The main service responsible for user management, generating discount codes, and tracking product click history.
2. **recommender:** The recommender service suggests personalized product recommendations based on the customer's purchase and view history.
3. **emailer:** The emailer service handles sending personalized birthday discount emails to eligible users.
4. **rainbow-queue:** Utilizing Bull and Redis, this service processes the message queue and sends emails/messages efficiently.
5. **corn:** This service is a cron job that continuously checks customers' birthdays and sends requests to the Rainbow Queue for processing.

Infrastructure Diagram

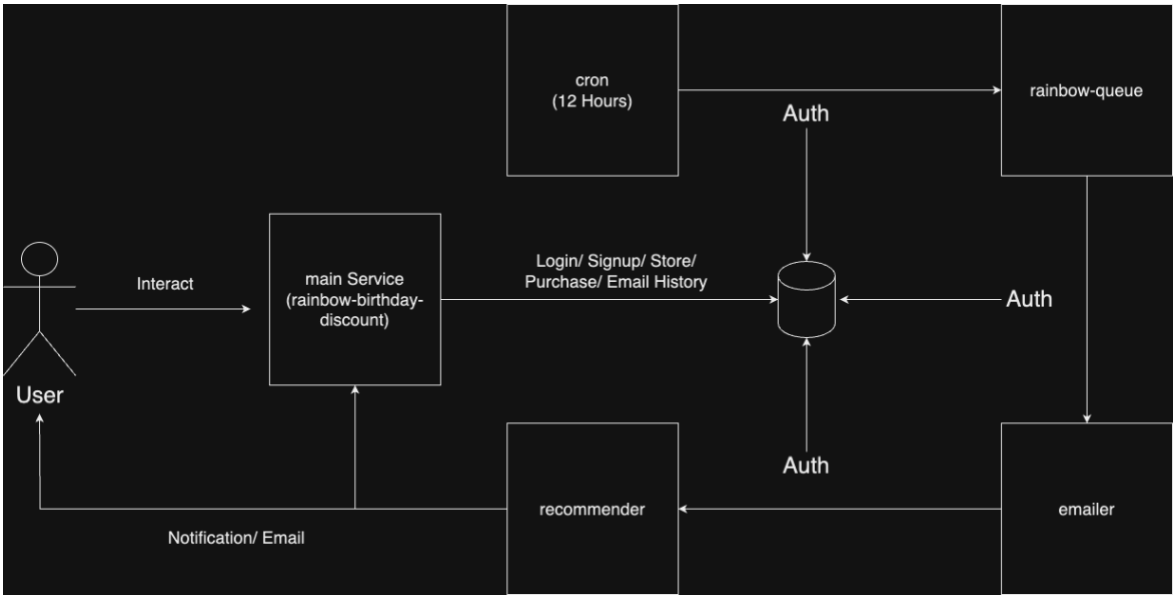

Fig: High Level Design For Rainbow Project

The infrastructure diagram illustrates the setup and communication between the different services. Each service runs in separate containers within Docker for easy deployment and scalability.

Technical Overview

Rainbow Birthday Discount System

The Rainbow Birthday Discount System is a NestJS-based application that allows you to send discounts to customers who have their upcoming birthdays in one week. The system consists of multiple services, each serving a specific functionality. The project uses Prisma as the ORM to interact with the PostgreSQL database.

Features

- **SignUp**: Allows customers to sign up and create an account.
- Login: Provides authentication for registered users.
- **JWT** : Authentication: Implements JSON Web Token (JWT) authentication for secure API access.
- **Guards**: Protects APIs using custom guards to ensure authorized access.
- **Recommender**: Recommends products to customers based on their purchase and view history.
- **Emailer**: Sends emails to customers, including discount codes and product recommendations.
- **Rainbow Queue**: Utilizes Bull as a message queue to send and track emails. Redis is used as the broker.
- **Cron Job (Corn**): Runs a cron job to continuously check customers' birthdays and sends requests to the Rainbow Queue.
- **PostgreSQL Database**: Uses PostgreSQL as the database to store customer information, products, discounts, etc.
- **Prisma ORM**: Leverages Prisma as the ORM to interact with the PostgreSQL database.

Getting Started

To run the project locally, follow these steps:

1. Clone the repository:
2. git clone <repository_url>
3. cd <repository_directory>
4. Install Dependencies

cd rainbow-birthday-discount
yarn install

cd ../recommender
yarn install

cd ../emailer
yarn install

cd ../rainbow-queue
yarn install

cd ../corn
yarn install

2. Set up the PostgreSQL database:

Make sure you have PostgreSQL installed and running. Create a database named "rainbow" with the appropriate username and password (default is "postgres:1234"). Set up the environment variables:

3. Create a .env file in the root directory and define the following environment variables:

DATABASE_URL="postgresql://postgres:1234@localhost:5434/rainbow?schema=public"

JWT_SECRET="super-secret"

EMAIL="takrim@user.com"

PASSWORD="1234"

4. Migrate the database using Prisma:

cd rainbow-birthday-discount

yarn prisma migrate dev

5. Build and run each service on different ports

cd rainbow-birthday-discount

yarn build

yarn start:dev

cd ../recommender

yarn build

yarn start:dev

cd ../emailer

yarn build

yarn start:dev

cd ../rainbow-queue

yarn build

yarn start:dev

cd ../corn

yarn build

yarn start:dev

Conclusion

The **Rainbow Birthday Discount** project demonstrates an analytical approach to provide a personalized discount experience to customers with upcoming birthdays. By utilizing a microservices architecture and message queue system, the project achieves modularity, scalability, and high performance. This documentation provides an overview of the project's architecture and infrastructure, ensuring a seamless experience for customers and efficient processing of birthday discount emails.