

CSE 586 Project 1 Phase 3 - Team 3

Topic - Food Nutrition Data Notification system with Pub/Sub Architecture

Public API Details:

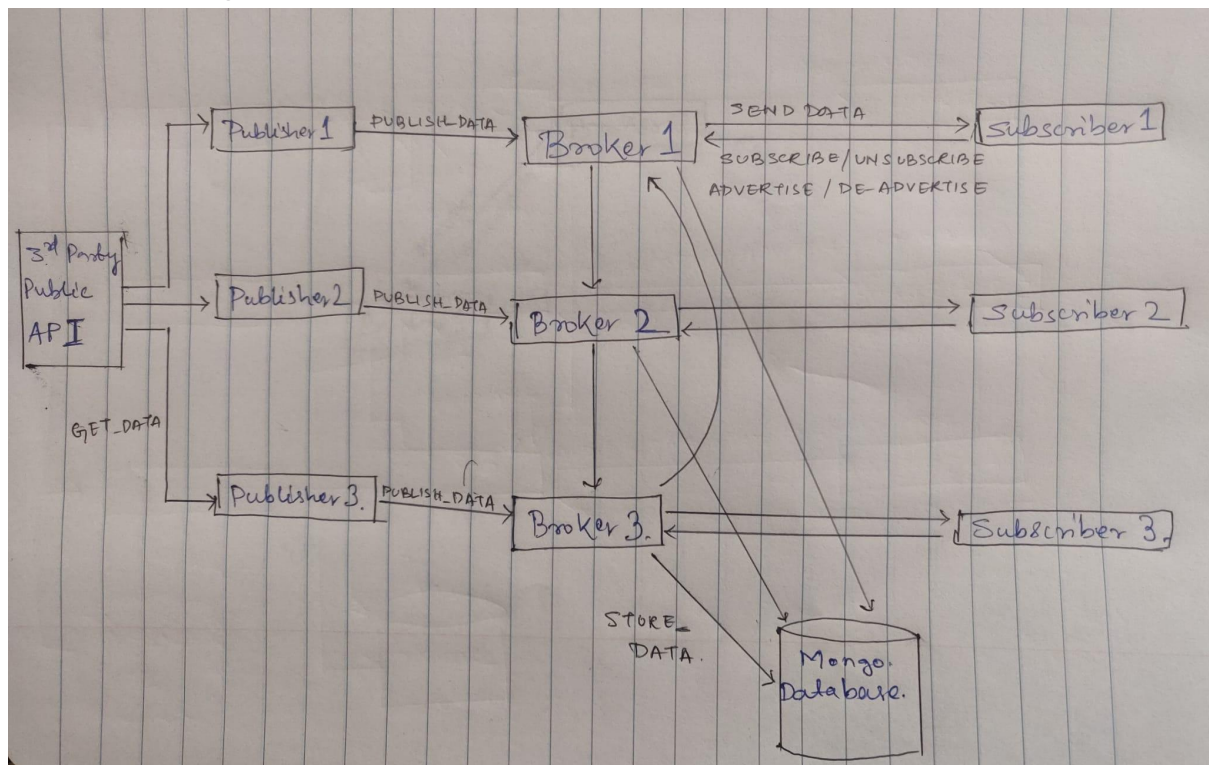
Document Link - [API Documentation](#)

Description - Natural language API to extract nutrition data from any text

Technologies Used:

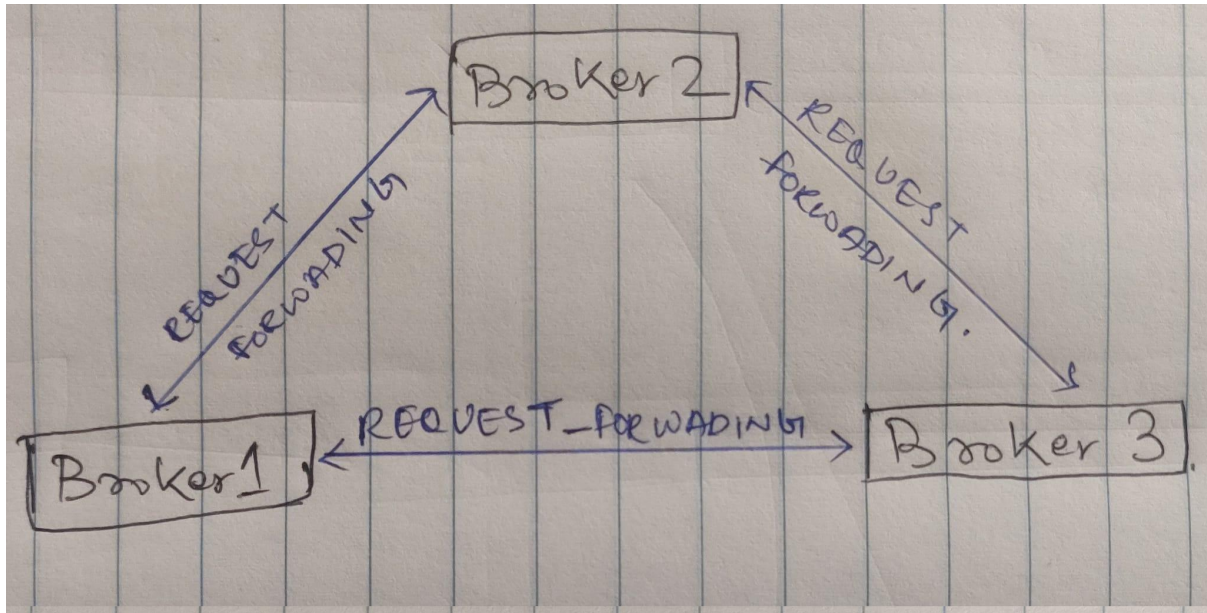
DataBase	MongoDB
Backend	NodeJS with Express
FrontEnd	NodeJS with ejs templates
Platform	Docker
Communication	Rest API

Architecture Diagram:



CSE 586 Project 1 Phase 3 - Team 3

Broker Network:



Topic-Publisher-Broker Relationship:

Publisher	Broker	Topic
Publisher - 1	Broker - 1	Chicken, Fish
Publisher - 2	Broker - 2	Beef
Publisher - 3	Broker - 3	Pork

Broker-Subscriber Relationship:

Broker	Subscriber
Broker - 1	Subscriber - 1
Broker - 2	Subscriber - 2
Broker - 3	Subscriber - 3

Explanation:

Publisher Functionalities:

- Periodically collects data from Public API for the specific topic
- On API Data collection publisher pushes the data to Broker

Broker Functionalities:

- Store data to DB on receiving data from publisher
- Forward the data to subscriber based on subscription and advertising preferences

CSE 586 Project 1 Phase 3 - Team 3

- Forward subscription request to neighbouring nodes if topic is out of its scope.
- Handle Subscribe/Unsubscribe/Advertise/DeAdvertise Request

Subscriber Functionalities:

- Display all the food details in UI
- Enables the user to subscribe or unsubscribe to a food topic
- Users can change preference to advertise and de-advertise
- Listens to subscriber for incoming data
- Refreshes the data automatically based on notification from the subscriber

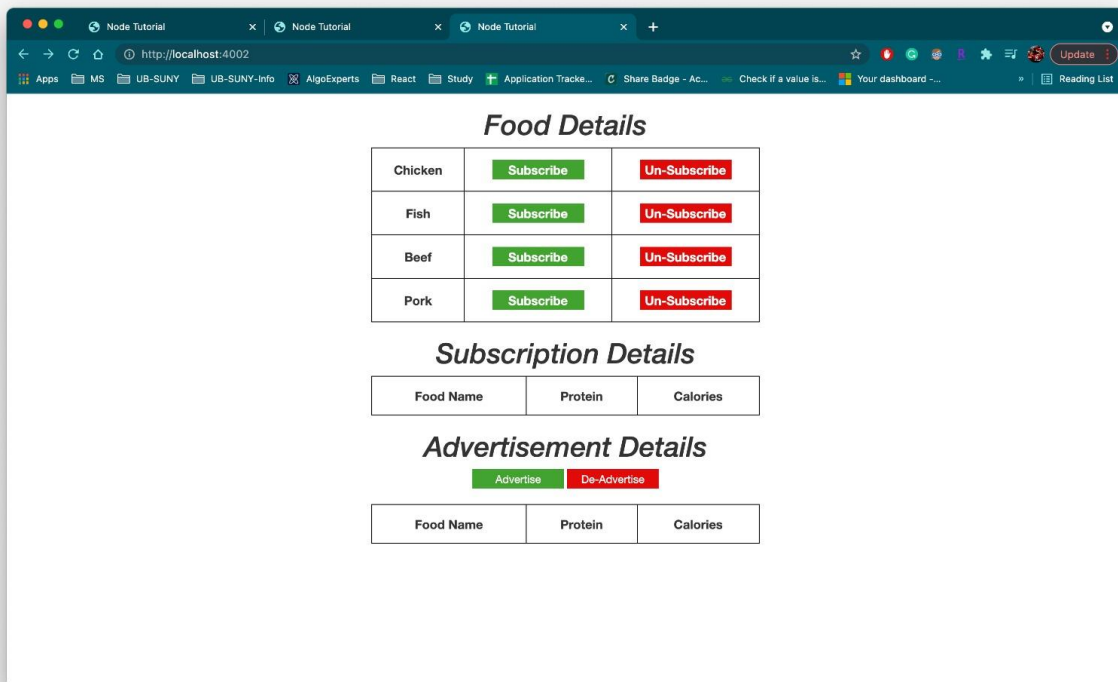
Contribution by team members:

Sivakumar Pasupathi(50366350)	Publisher functionalities, Subscriber functionalities
Parthiban Rajendran(50415960)	Broker functionalities, Docker Platform
Common Handling	Request Forwarding in broker, Rest API Communication between Publisher, Broker and Subscriber

ScreenShots:

User Interface

1) Basic View



CSE 586 Project 1 Phase 3 - Team 3

2) View After Subscribing to a topic

The screenshot shows a web application running on a browser at `http://localhost:4002`. The application has three main sections:

Food Details

Chicken	<button>Subscribe</button>	<button>Un-Subscribe</button>
Fish	<button>Subscribe</button>	<button>Un-Subscribe</button>
Beef	<button>Subscribe</button>	<button>Un-Subscribe</button>
Pork	<button>Subscribe</button>	<button>Un-Subscribe</button>

Subscription Details

Food Name	Protein	Calories
chicken	23.7	222.6

Advertisement Details

Advertise De-Advertise

Food Name	Protein	Calories
-----------	---------	----------

3) View with subscription with advertising enabled

The screenshot shows the same web application as before, but with the 'Advertisement Details' section expanded to show four rows of data:

Food Details

Chicken	<button>Subscribe</button>	<button>Un-Subscribe</button>
Fish	<button>Subscribe</button>	<button>Un-Subscribe</button>
Beef	<button>Subscribe</button>	<button>Un-Subscribe</button>
Pork	<button>Subscribe</button>	<button>Un-Subscribe</button>

Subscription Details

Food Name	Protein	Calories
chicken	23.7	222.6

Advertisement Details

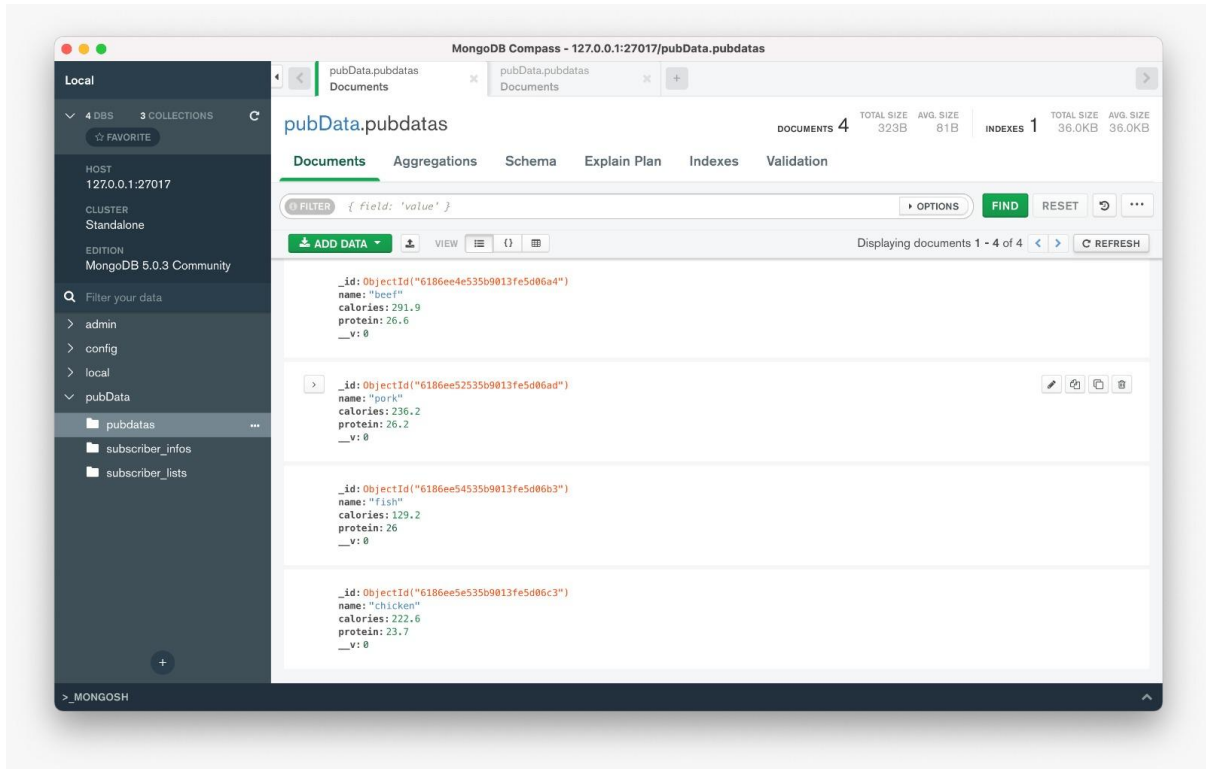
Advertise De-Advertise

Food Name	Protein	Calories
chicken	23.7	222.6
fish	26	129.2
beef	26.6	291.9
pork	26.2	236.2

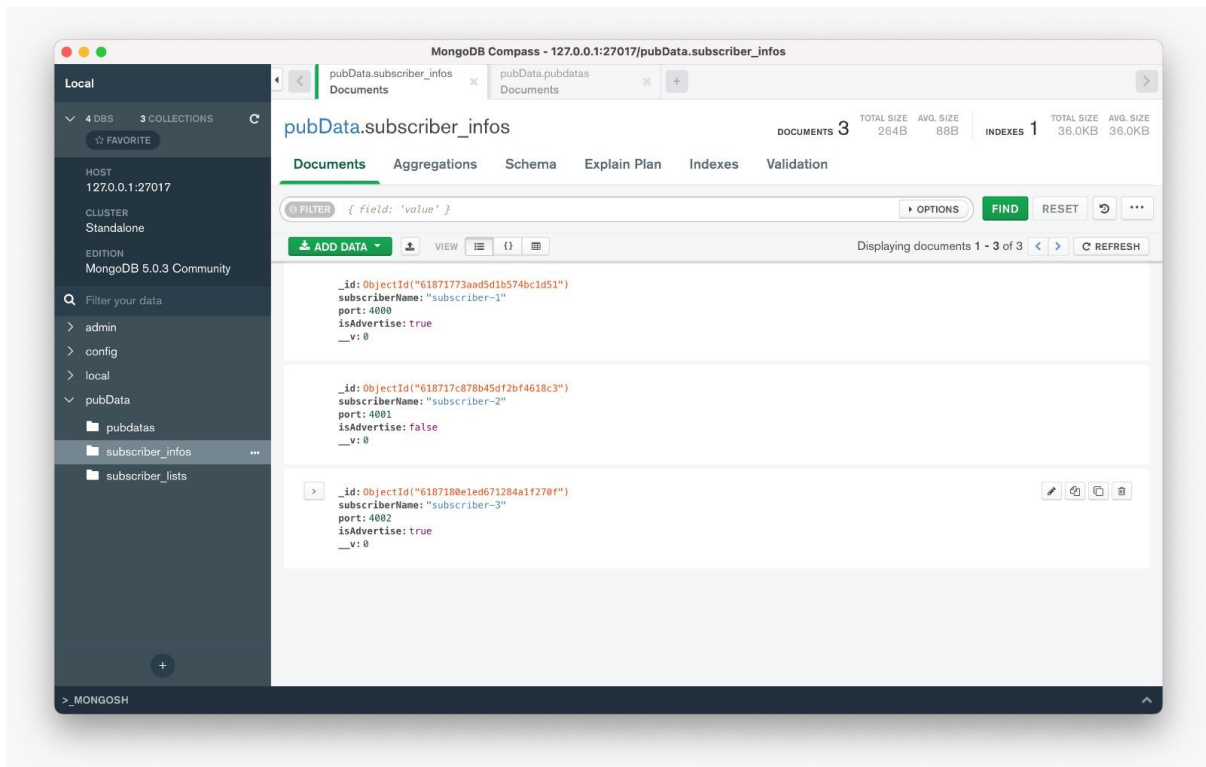
CSE 586 Project 1 Phase 3 - Team 3

Database Schema

1) Published Data Collection schema



2) Subscriber Info collection schema



CSE 586 Project 1 Phase 3 - Team 3

3) Subscriber List Collection schema

