

MASTER OF TECHNOLOGY

INTELLIGENT DIET ASSISTANT SYSTEM

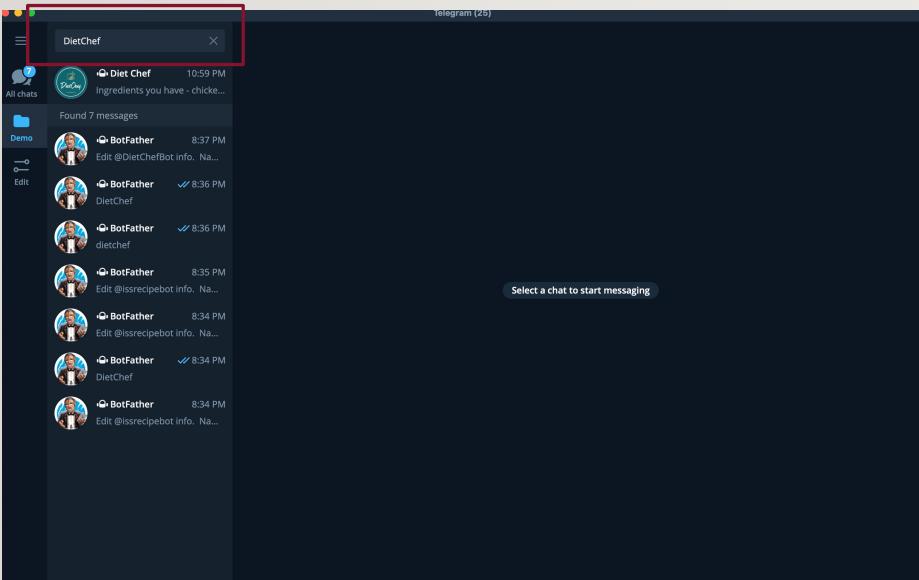
TEAM MEMBERS

- Chen Hao
- Chiu Man Shan
- Kuch Swee Cheng
- Zhao Lutong

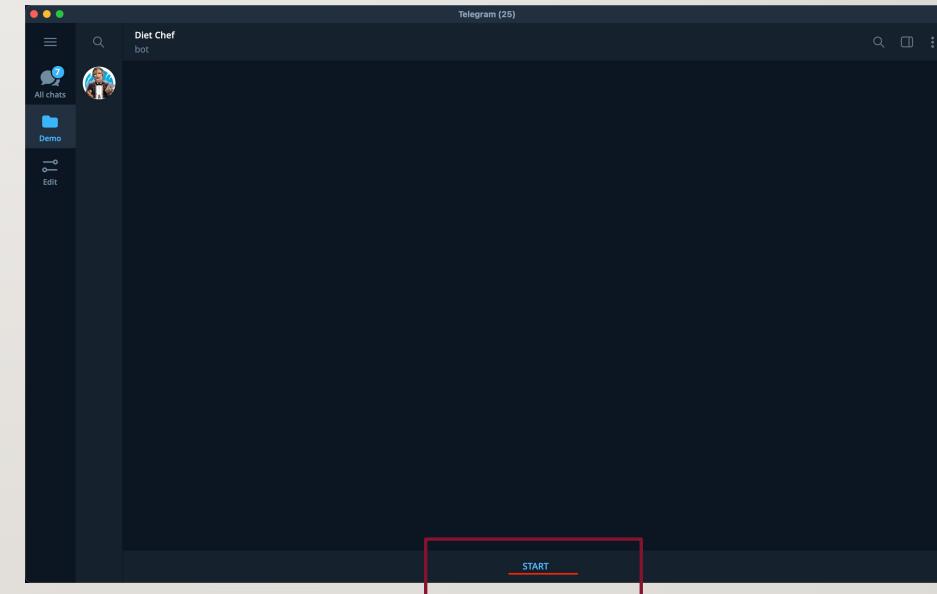
USER GUIDE

- User Interface
 - We are using telegram bot as our UI and you can ask the bot questions using commands like “/estimate_calories xxx” and it will reply you the necessary information.

USER GUIDE (START TALKING TO THE BOT)



Search “DietChef” in your telegram application

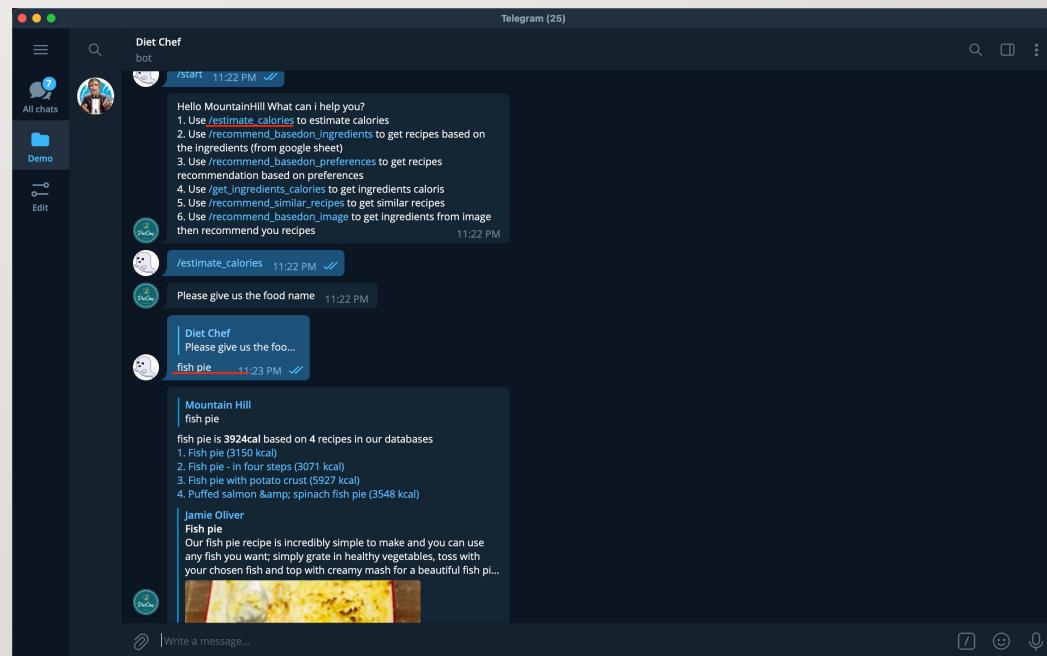


Click start to start the conversation

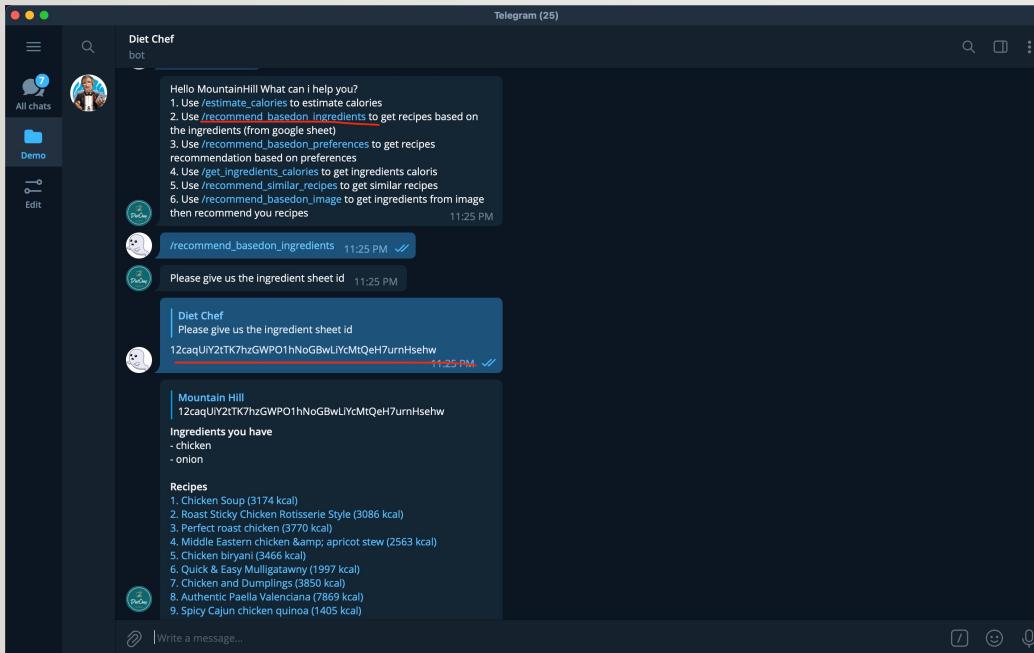
USER GUIDE(ESTIMATE CALORIES)

Feature I

Click “estimate_calories” and the bot will force you to reply with the food name and it will give you back the estimated calories based on the recipes we have



USER GUIDE(RECOMMEND ON INGREDIENTS)



Feature 2

Click

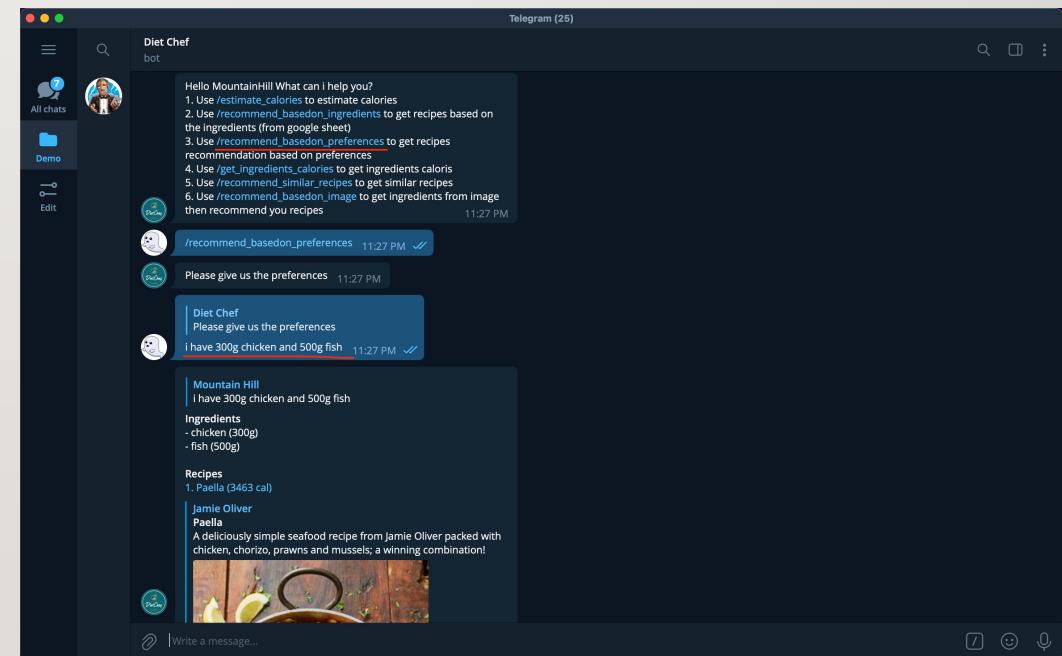
"recommend_basedon_ingredients"
and the bot will force you to give it a
google sheet id and it will recommend
recipes based on the recipes you
provided in the google sheet

USER GUIDE(RECOMMEND ON PREFERENCES)

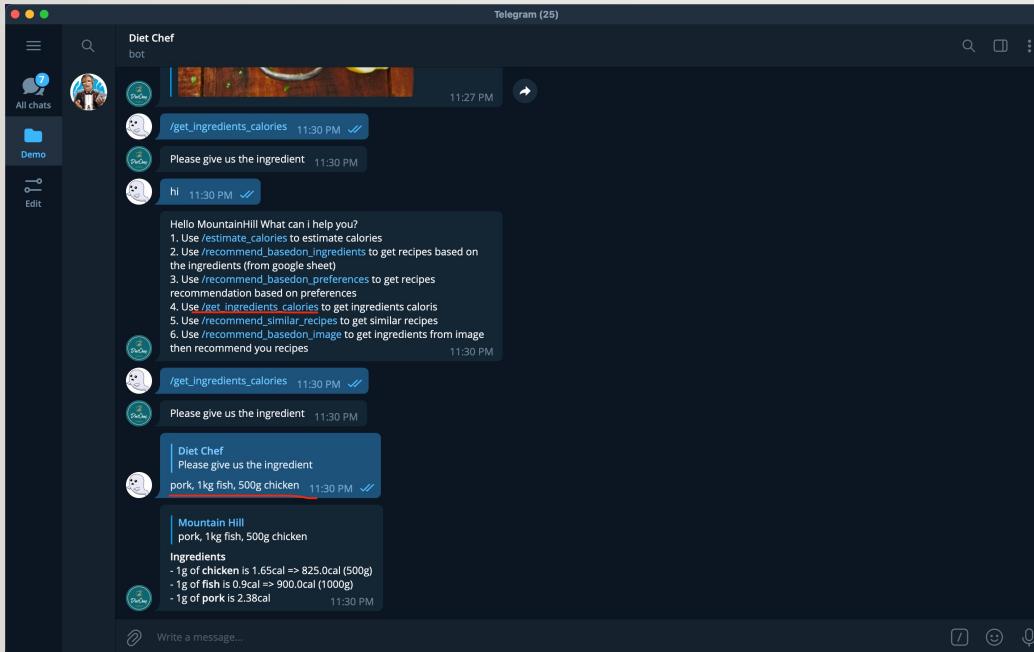
Feature 3

Click

“recommend_basedon_preferences”
and just reply your preferences and
the bot will give you recipes
recommendation



USER GUIDE(GET INGREDIENTS DETAIL)



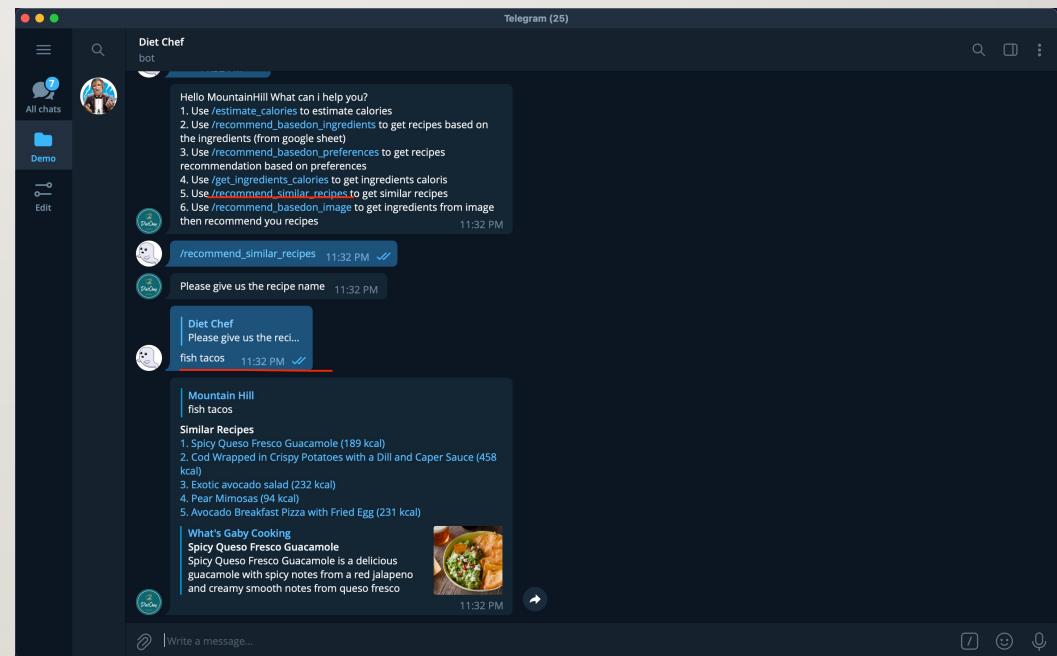
Feature 4

Click “`get_ingredients_calories`” and give the bot the ingredients and the weight that you want to check and it will reply you the corresponding calories

USER GUIDE(RECOMMEND SIMILAR RECIPES)

Feature 5

“recommend_similar_recipes” and just give the recipe name and the bot will give you the most similar recipes



END OF USER GUIDE

INSTALLATION GUIDE (REQUIREMENTS)

- AWS account
 - Free trial account can be registered [here](#)
- Elasticsearch (version 8)
 - Free trial account can be registered [here](#)
 - You can also start a [docker image](#) to test it locally
- Telegram bot
 - You can create a new bot by talking to [BotFather](#)

INSTALLATION GUIDE (PACKAGES)

- Pandas
- Nltk
- Python-dotenv
- Spacy (en_core_web_sm model to do text preprocessing)
- Requests
- Numpy
- Gensim (word2vec's cbow to test model)
- sklearn

INSTALLATION GUIDE (ENVIRONMENT VARIABLES)

- Before you run the script, you have to create .env file under the root directory with the following variables
- ELASTIC_HOST (Get from Elasticsearch Cluster)
 - Sample: <https://xxxxx:9243>
- ELASTIC_USER (Get from Elasticsearch Cluster)
 - Sample: elastic
- ELASTIC_PASS (Get from Elasticsearch Cluster)

INSTALLATION GUIDE (PREPROCESSING)

- Run preprocess/01_fetch_data.ipynb
 - Pull data from forkify and nutrition 5k
 - Extract ingredients and unit from the source data
- Run preprocess/02_recipe_similarity.ipynb
 - Calculate similarity between recipes
- Run preprocess/03_preprocessdata.ipynb
 - Prepare data for tf-idf text search later
 - Group all the data and push to elasticsearch

INSTALLATION GUIDE (BOT, ENV FILE)

Create .env file under SystemCode/telegrambot folder as following or refer to the .env.sample file

- **BOT_TOKEN** (Get from Bot Father)
- **GOOGLE_TOKEN** (Get form google console)
 - It is the token for the bot to read google sheet
- **ELASTIC_CLOUDID** (Get from Elasticsearch)
- **ELASTIC_USER** (Get from Elasticsearch)
- **ELASTIC_PASS** (Get from Elasticsearch)
- **RECIPE_INDEX**
 - It is the index name that stores the recipe data
- **INGREDIENT_INDEX**
 - It is the index name that stores the ingredient data

INSTALLATION GUIDE (SYSTEM PACKAGES)

- Install [NodeJS](#)
- “npm install serverless -g” this is used to deploy lambda later

INSTALLATION GUIDE (DEPLOY TO AWS)

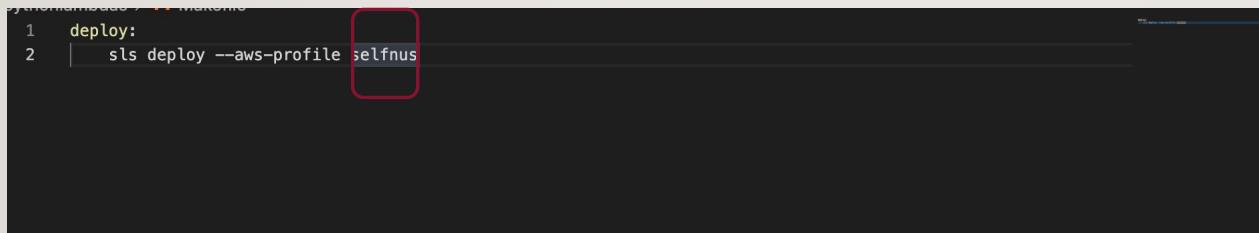
- You need to update this part (SystemCode/telegrambot/package.json) to your own AWS credential

```
11 "scripts": {  
12   "test": "echo \"Error: no test specified\" && exit 1",  
13   "start": "node --inspect serverless offline",  
14   "deploy": "(cd src/layer/nodejs && npm install --production) && npx sls deploy --aws-profile selfnus --region ap-southeast-1",  
15   "build": "npm install",  
16   "install-layer": "node copy-layer-packages.js && (cd src/layer/nodejs && npm install --production)"  
17 },  
18     You, last month * init ...  
19   "scripts": {  
20     "start": "node --inspect serverless offline",  
21     "deploy": "(cd src/layer/nodejs && npm install --production) && npx sls deploy --aws-profile selfnus --region ap-southeast-1",  
22     "build": "npm install",  
23     "install-layer": "node copy-layer-packages.js && (cd src/layer/nodejs && npm install --production)"  
24   }  
25 }
```

- Under SystemCode/telegrambot
 - Run “npm run install” to install all necessary packages
 - Run “npm run deploy” to deploy the bot backend to your AWS account

INSTALLATION GUIDE (DEPLOY TO AWS, PYTHON)

- You need to update this part (SystemCode/pythonlambdas/Makefile) to your own AWS credential



```
pythonlambdas/7> makefile
1 deploy:
2     sls deploy --aws-profile selfnus
```

A screenshot of a terminal window showing a portion of a makefile. The file contains two lines: 'deploy:' and 'sls deploy --aws-profile selfnus'. A red box highlights the 'selfnus' part of the AWS profile argument.

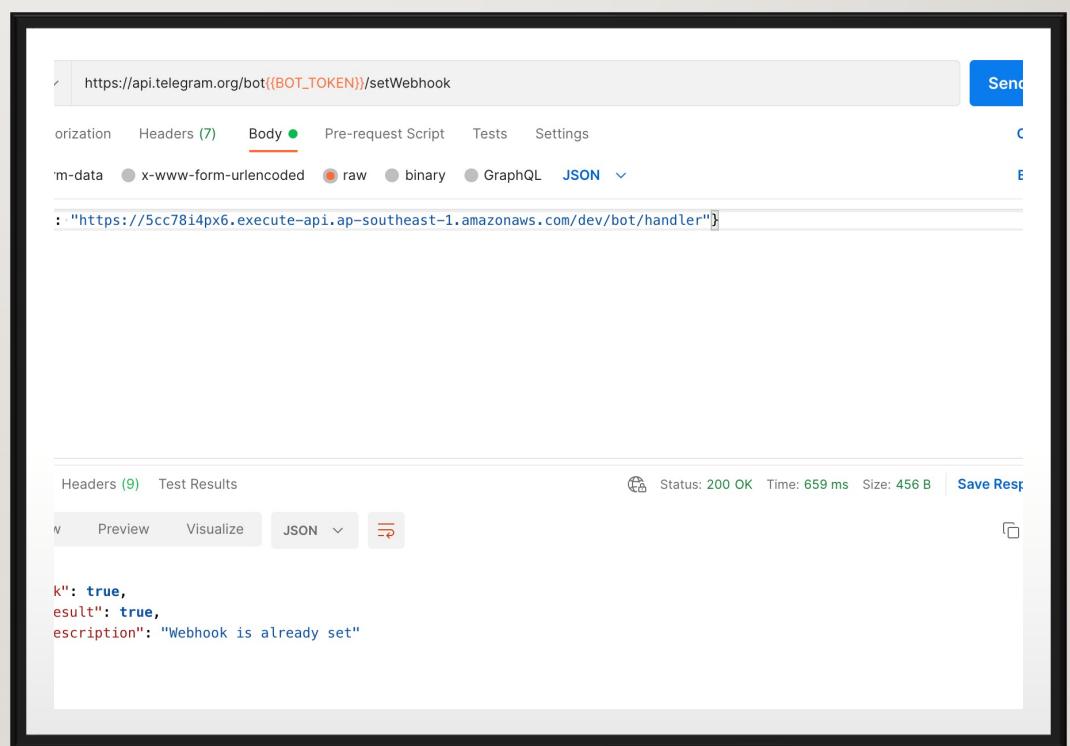
- Under SystemCode/pythonlambdas
 - Run “make deploy”

INSTALLATION GUIDE (ENABLE WEBHOOK)

- Last time is to enable the telegram webhook by calling the api.
 - POST

https://api.telegram.org/bot{{BOT_TOKEN}}/setWebhook

 - BOT_TOKEN is from botfather
 - BODY: {"url": ""} where the url is the deployed endpoint from AWS



END