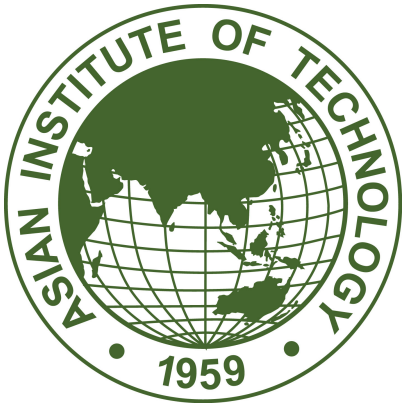
User and Installation Manual

CocoTrace: Product Traceability using QR Code and Chatbot

AT70.19 Software Development and Quality Improvement

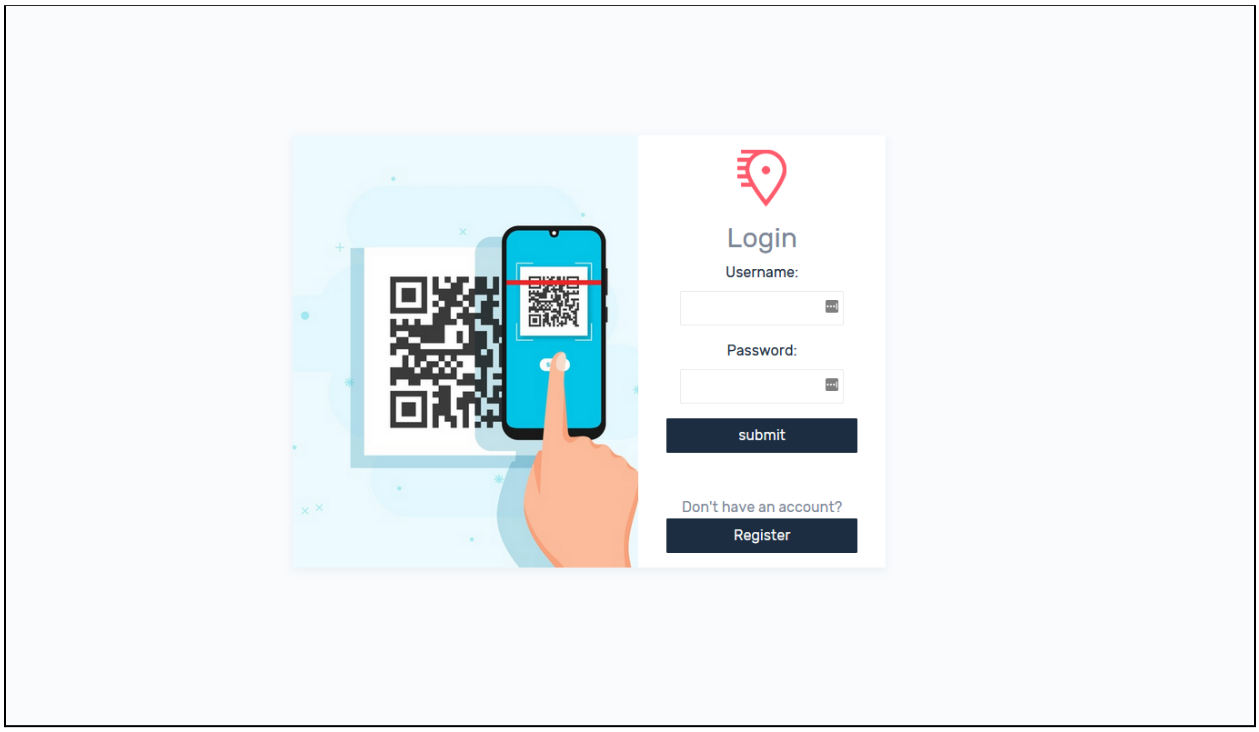


Design and Development Team

Htoo Lwin - st120832
Cedric Le Mercier - st122004
Titouan Le Mao - st122014
Urusha Rajkarnikar - st121263

1. Using the Web Application

Visit https://cocotrace.herokuapp.com. It can take a while to open at first, because Heroku free dynos on which the app is running can take a few seconds to a minute to start.



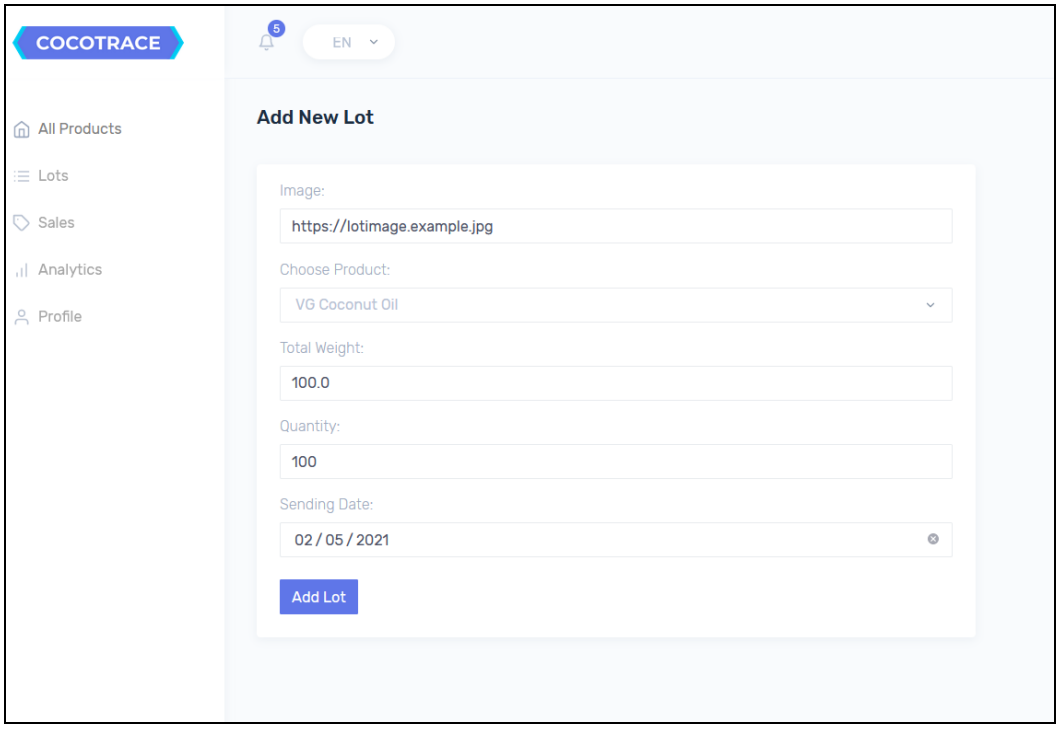
1.1 Log in as an administrator (or manufacturer/producer)

With these credentials:

```
Username: administrator
Password: administrator
```

1.2 Creating a new lot

The manufacturer can create new lots and add a product to it. A lot can only have one product. Browse to **“Lots”** in the left side navigation menu and click on **“Add New Lot”** on the next page.



Fill the fields:

- **Image:** Take the link of an image for the lot (JPG/PNG), make sure it is an actual image link with .jpg/.png at the end.
- **Product:** The product the lot contains
- **Total Weight:** Total weight of all products combined in the lot
- **Quantity:** Total number of individual products items in the lot
- **Sending Date:** The date the manufacturer ships the lot

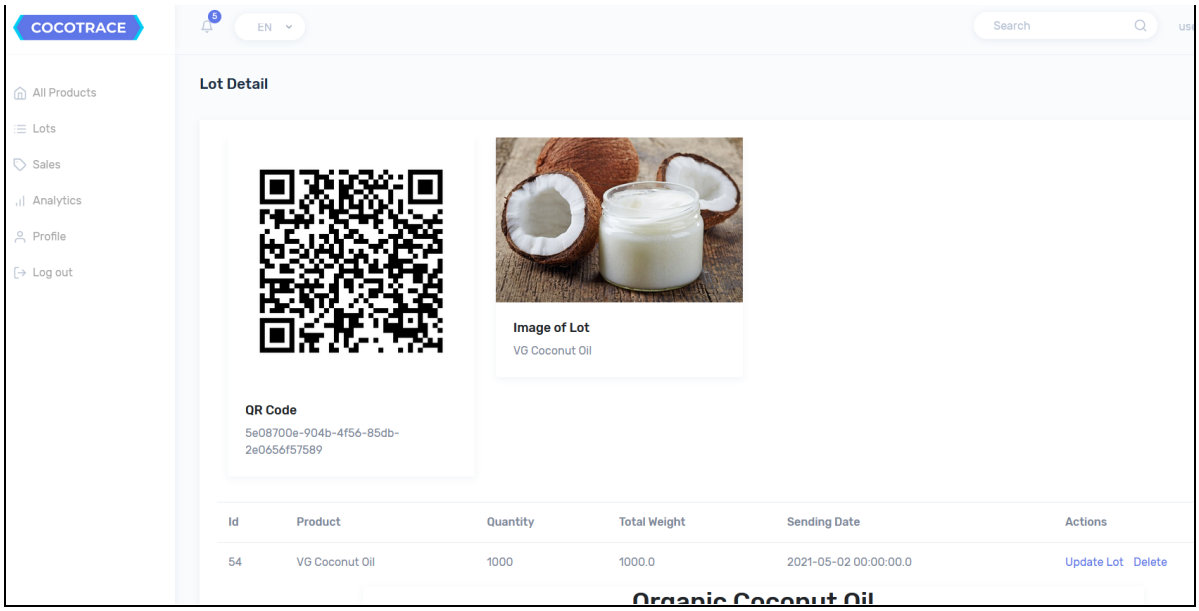
1.3 Lot Detail Page

After creating the lot, you will be redirected to the lot detail page: It contains all information related to the lot including:

- **Generated QR Code:** This QR code links to an URL with the ID of the lot with the following format. On scanning, this link either get JSON data about the lot or the lot detail webpage, depending whether you scanned from the **LINE** app or not:

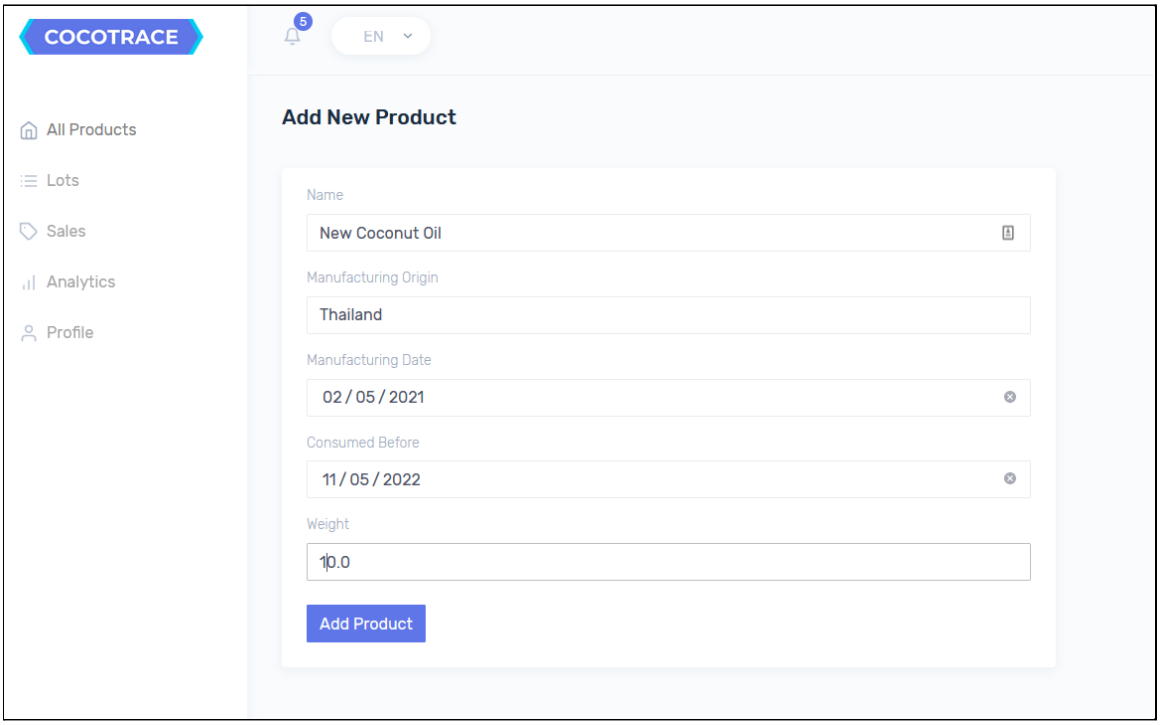
```
https://{WEBSITE_NAME}/redirectToWebOrLine?id={ID_OF_THE_LOT}&qrId={QR_CODE_ID}
```

- **QR Code Id:** Can also be used as a lot ID, this will be printed along with QR Codes on individual products in the lot



1.4 Creating new products

You can also create new products to add to a lot. For this, visit **ALL PRODUCTS** in the left side navigation, and click **Add New Product** on the following screen. Enter the details and click **Add Product**



1.5 Retrieve product or lot data

Through website

Scan the QR code provided above with your phone and it will redirect you to the appropriate webpage. Since a lot contains only one type of product, scanning the lot QR code will also get the product data that the lot contains.

Through LINE Chatbot

Follow the guide in the next section.

2. Using the LINE Chatbot

2.1 Adding the Chatbot in LINE App

Scan the QR Code below to add

2.2 Commands

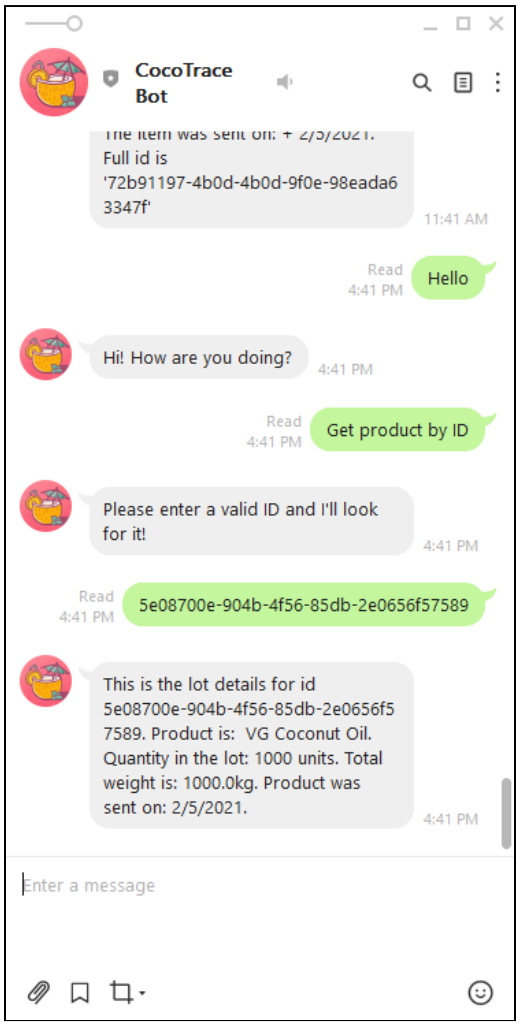
Currently, the chatbot supports only 3 commands

- **Getting help or suggestion of commands**
    - Just type “Help” or “Commands” or “What can you do”
- **Retrieving lot by ID**
- **Retrieving lot by scanning QR Code**

2.3 Retrieving lot or product data

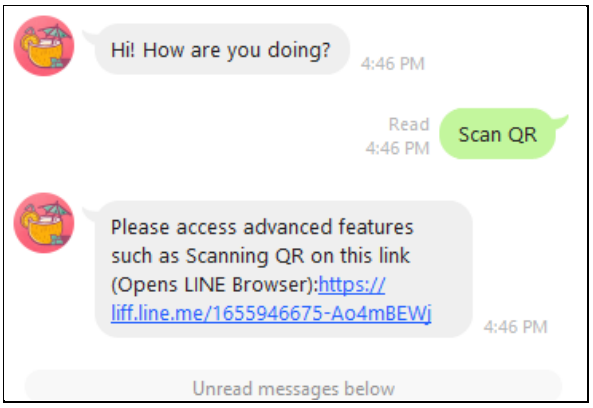
By Entering Lot Qr Id

Enter "**Get product by ID"** or just "**Get product"**
The bot will prompt you to enter the **QR Code Id** that displayed below the QR code in **Section 1.3**

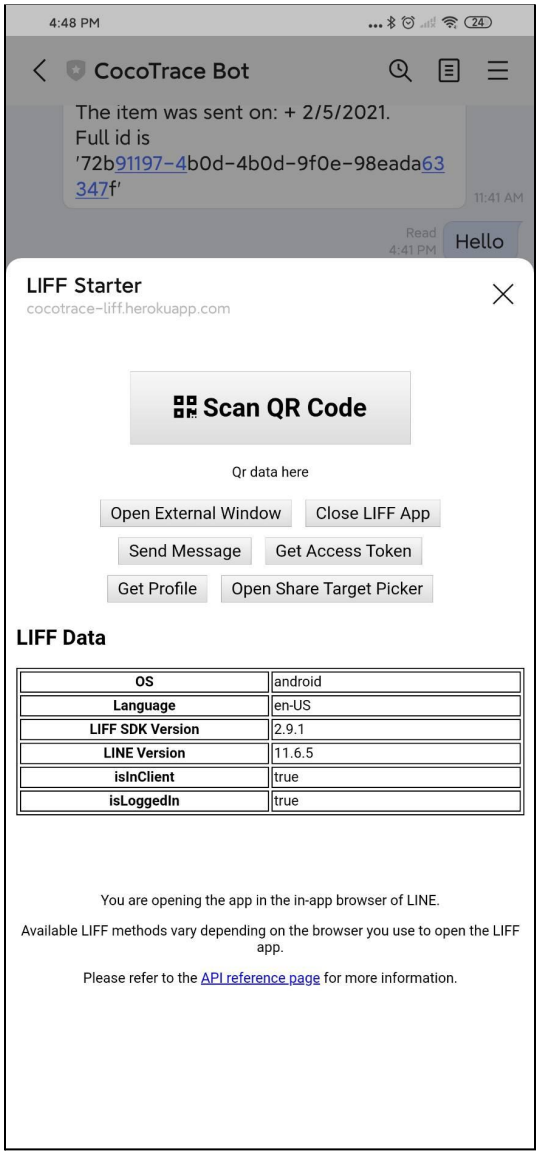


By Scanning a QR Code

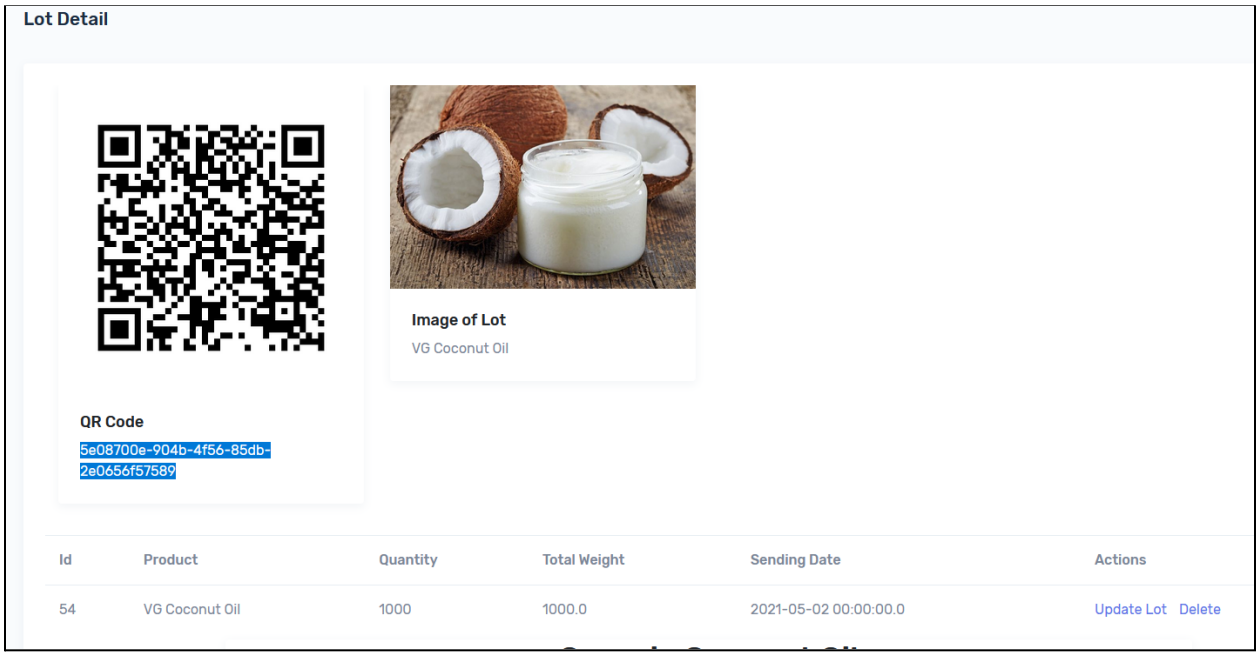
Type **"Scan QR code"** or just **"Scan QR".**
The bot will send you back the link of a LIFF app from which you can do several actions, including scanning QR Code



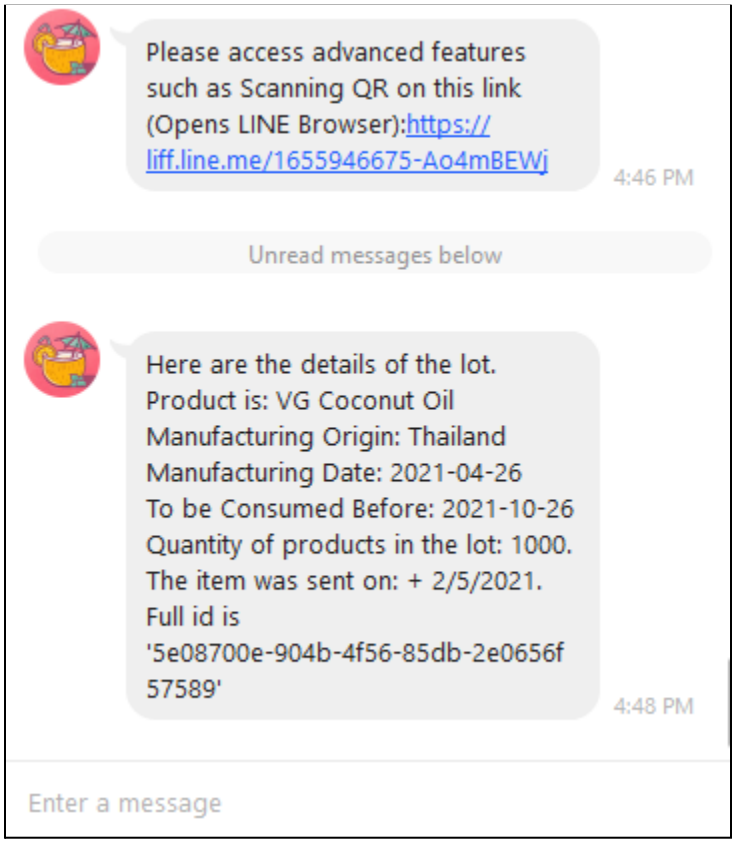
This opens the LINE browser with LIFF app which is currently hosted on **https://cocotrace-liff.herokuapp.com/**



Scan the lot QR code and wait for the bot to retrieve data.

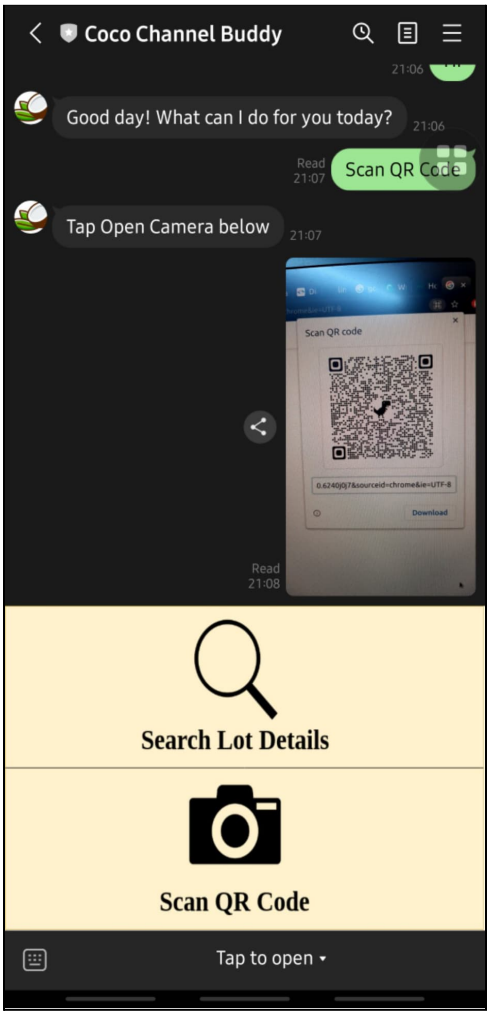


NOTE: *This can take several seconds for the cocotrace LIFF App to start up on Heroku. If nothing happens, scan the QR code again*



2.4 Scanning QR Code through Rich Menu

This is another chatbot that we were trying but we could not fully implement. By tapping an area on the rich menu we can prompt the user to take or select a photo but we could not process the image further through DialogFlow. A video demo will also be submitted for this method.



3. Setting up the a local environment for development

3.1 Set up

Clone the git repository https://github.com/cocostuff/cocotrace

```
git clone https://github.com/cocostuff/cocotrace.git
```

3.2 Download and install PostgreSQL 13

Go to https://www.postgresql.org/download/ and follow the instructions for installing. By default, install on port **5432.** In the components installation screen, make sure to tick **"pgAdmin 4"** to install the admin user interface for managing PostgreSQL.

Create an admin user for database in pgAdmin 4

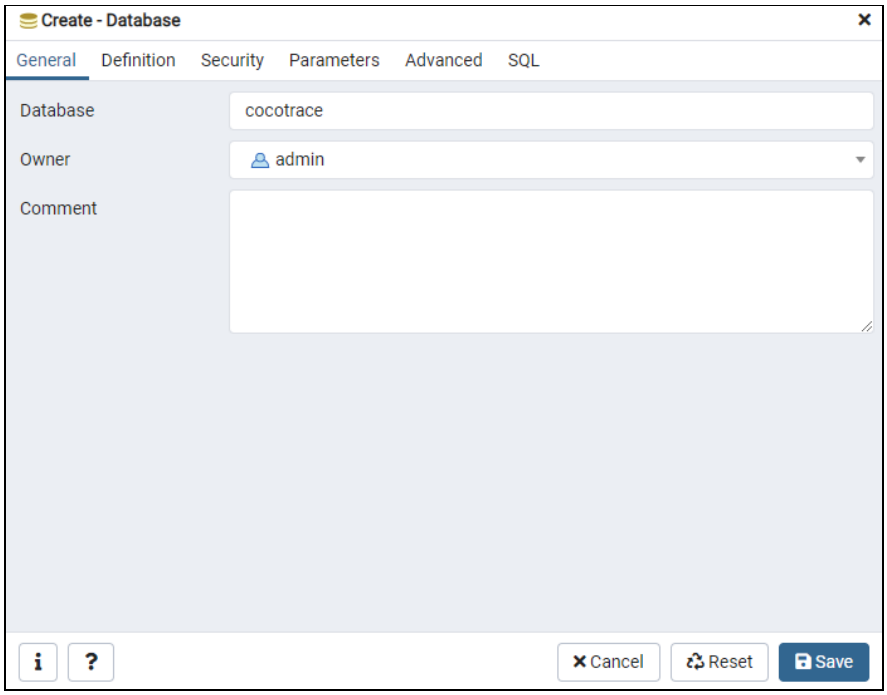
By default, the application is configured with the following database admin credentials, either create a user in pgAdmin with these credentials or change ***application.properties*** configuration in ./src/main/resources/

```
spring.datasource.username=admin
spring.datasource.password=admin
```

Note: *Windows users can experience an 'authentication type 10 error' during first time setup. Check the troubleshooting section for more information.*

Create a table for cocotrace in pgAdmin 4

Create a database named **cocotrace** and set the newly created admin user as the Owner



3.3 Run the app

Run ***CocotraceApplication.java*** as the main class. During the first run, it should create 6 tables within the cocotrace database and populate the tables with the queries contained in **src/main/resources/data.sql.**
Visit the application at **http://localhost:8082** and login as an administrator with the default credentials

```
Username: administrator
Password: administrator
```

If it doesn't work, there was possibly an error with the queries populating table. Check the troubleshooting section "*Populate the tables with dummy data manually with pgAdmin 4*"

4. Deploying to production on Heroku

To deploy Spring Boot applications to heroku, follow this guide:
https://devcenter.heroku.com/articles/deploying-spring-boot-apps-to-heroku
NOTE: When publishing the Spring Boot app on Heroku, there is no need to create a database as it will be created automatically

4.1 Populate Heroku database

In some cases, the tables will be created empty and you will need to run the queries manually to add default and dummy data. Get the queries in the *Troubleshooting section*.
To connect to Heroku database from CLI, login to Heroku via CLI and run:

```
heroku pg:psql
```

Then copy paste the SQL queries one by one.
More info on Heroku PostgreSQL and CLI:
https://devcenter.heroku.com/articles/heroku-postgresql#using-the-cli

4.2 Setup the environment variables in Heroku

Set up the environment variables for the app to work
For only the webapp:

- **APP_URL:** Put your heroku app URL here (only used to create QR Codes links)
- **DATABASE_URL:** The heroku database url, beginning with postgres://

For LINE Chatbot to work only (If creating your own Dialogflow and LINE chatbot)

- **CHANNEL_TOKEN:** Put the Chatbot MESSAGING API Line channel token here (necessitates a LINE Developer account and create a Chatbot using Messaging API)
- **LIFF_URL:** The LIFF App URL (Check Deploying the LIFF App section).

5. Deploying the LIFF App

Current url for the LIFF Application **https://cocotrace-liff.herokuapp.com/**
Note: Some functionalities are only available in LINE application, such as scanning QR code)

5.1 Overview

LINE Front-end Framework (LIFF) is a platform for web apps provided by LINE. The web apps running on this platform are called LIFF apps. The LIFF app is what allows the chatbot to access advanced features, such as scanning QR code or getting user profile information.
Clone the repository

```
git clone https://github.com/cocostuff/cocotrace-liff
# then run to install dependencies
npm install
```

5.2 Deploy the app on Heroku

Follow the instructions on deploying node.js apps
https://devcenter.heroku.com/articles/deploying-nodejs

Note: To make the LIFF app work with your chatbot, it requires creating a login channel and LIFF application in your bot provider.

6. Troubleshooting

5.1 (Windows) Authentication type 10 error

Follow the instructions here for fixing this issue:
https://stackoverflow.com/questions/64210167/unable-to-connect-to-postgres-db-due-to-the-authentication-type-10-is-not-suppor

5.2 Populate tables with default & dummy data manually

Open pgAdmin, click on the **cocotrace** database newly created an paste those queries

```
insert into role (name) values ('ROLE_ADMIN');
insert into role (name) values ('ROLE_USER' );

--default login: user:administrator pw: administrator
insert into user_account (active, email, password, username, role_id) values (true, 'test@test.com',
'$2a$10$qQn/YI3iGA/iOaNqxqQ5HeYKIOyUEcemGLGuKFyyCWaIgfBDWEx9a', 'administrator', 1);
insert into product (consumed_before, manufacturing_date, manufacturing_origin, name, weight, user_id)
values ('2021-04-21', '2021-04-14', 'Thai', 'Oil Coconut 100ML', 100.0, 1);
insert into product (consumed_before, manufacturing_date, manufacturing_origin, name, weight, user_id)
values ('2021-04-21', '2021-04-14', 'Taiwan', 'Oil Coconut 200ML', 200.0, 1);
insert into lot (global_weight, qr_code_id, quantity, sending_date, product_id, user_id) values (250.0,
'a882ccc44000409fa22941c1f497c185', 50, '2021-04-07', 1, 1);
```