ข้อที่2 มีการเตรียมการโดยใช้ docker

โดยแยกเป็น Backend Frontend

สำหรับ Backend สร้างไฟล์

# Docker for Python API

FROM python:3.9-slim

WORKDIR /app

COPY requirements.txt .

RUN pip install --no-cache-dir -r requirements.txt

COPY . .

CMD ["python", "app.py"]

สำหรับ Frontend สร้างไฟล์

# Dockerfile for Vue.js

FROM node:14

WORKDIR /app

COPY package.json package-lock.json ./

RUN npm install

COPY . .

RUN npm run build

EXPOSE 8080

CMD ["npx", "http-server", "dist", "-p", "8080"]

ใน yml

version: '3.8'

services:

api:

build:

context: ./api

ports:

- "5000:5000" # port ที่ใช้รันเซิร์ฟเวอร์

frontend:

build:

context: ./frontend

ports:

- "8080:80" #port ตั้งตามที่ต้องการ

หากมีการนำไปใช้ในสภาพแวดล้อมที่อื่นสามารถใช้

docker pull …name project…

docker run -d -p host-port:container-port …name project…

ก็จะสามารถใช้โปรเจคได้อย่างง่ายดาย

ข้อที่ 3

<https://www.figma.com/design/QZVnKX8oggwk6V249wCen6/Untitled?node-id=0-1&t=Y7BD3qfrkHK21QvF-0>

เมนูหลัก

Queue ดูรายละเอียดต่างๆตามที่ออกแบบ สามารถเลือกแบบต่างๆตามที่ select

สามารถดูสถานการณ์ส่งได้ดังนี้

สีแดง มีข้อผิดพลาด หรือไม่สำเร็จ

สีเขียว การขนส่งสำเร็จ

สีเหลือง การขนส่งกำลังดำเนินการ

โดยสามารถเข้าไปดูในรายละเอียด ที่มากขึ้นโดยการคลิกเข้าไปที่ card ที่ต้องการที่จะดู ลบ แก้ไข ข้อมูล

การเพิ่ม คิวคลิกที่ปุ่ม+

Truck ดูรายละเอียดของรถต่างๆเข้ามาส่งของ เพิ่ม ลบ แก้ไข ข้อมูลของรถเป็นต้น

Warehouse ดูของที่ทำการส่งมีจำนวนเท่าไร่ หรือดูเป็นรอบการส่งมาส่งจำนวนเท่าใด หรือในแต่ละรอบมีรอบการส่งเท่าไร่เป็นต้น