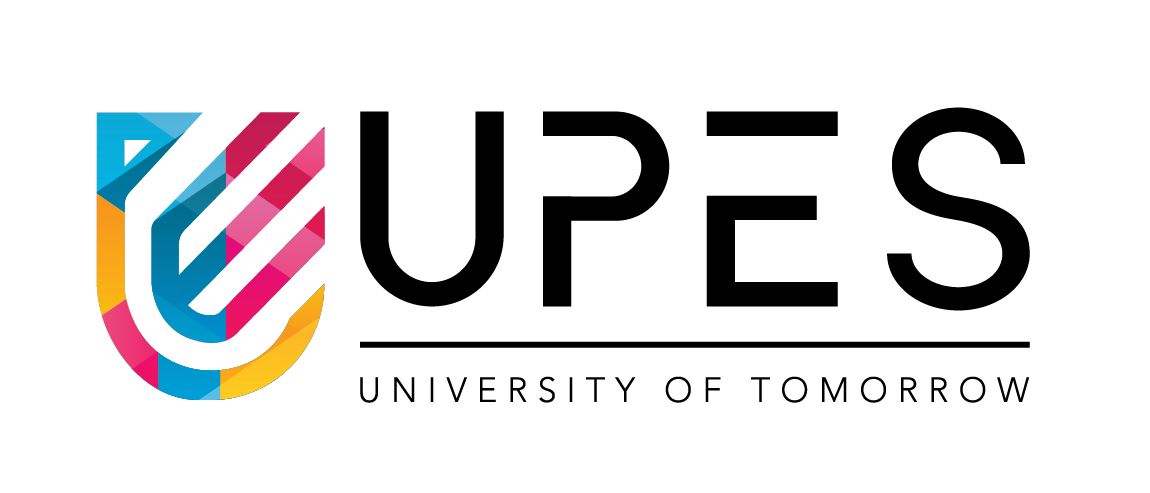
****

**DOCKER ASSIGNMENT**

**Submitted by:**

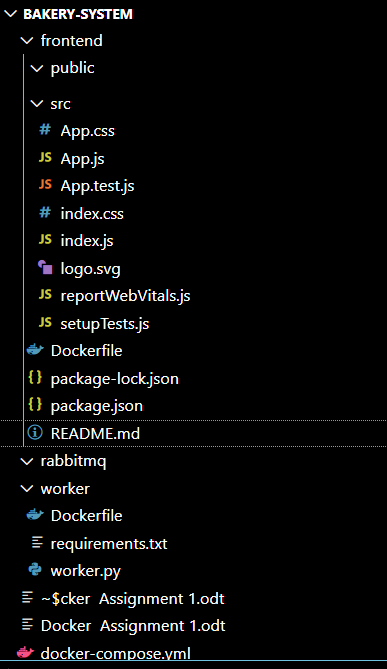
**KHUSHI PANDEY**

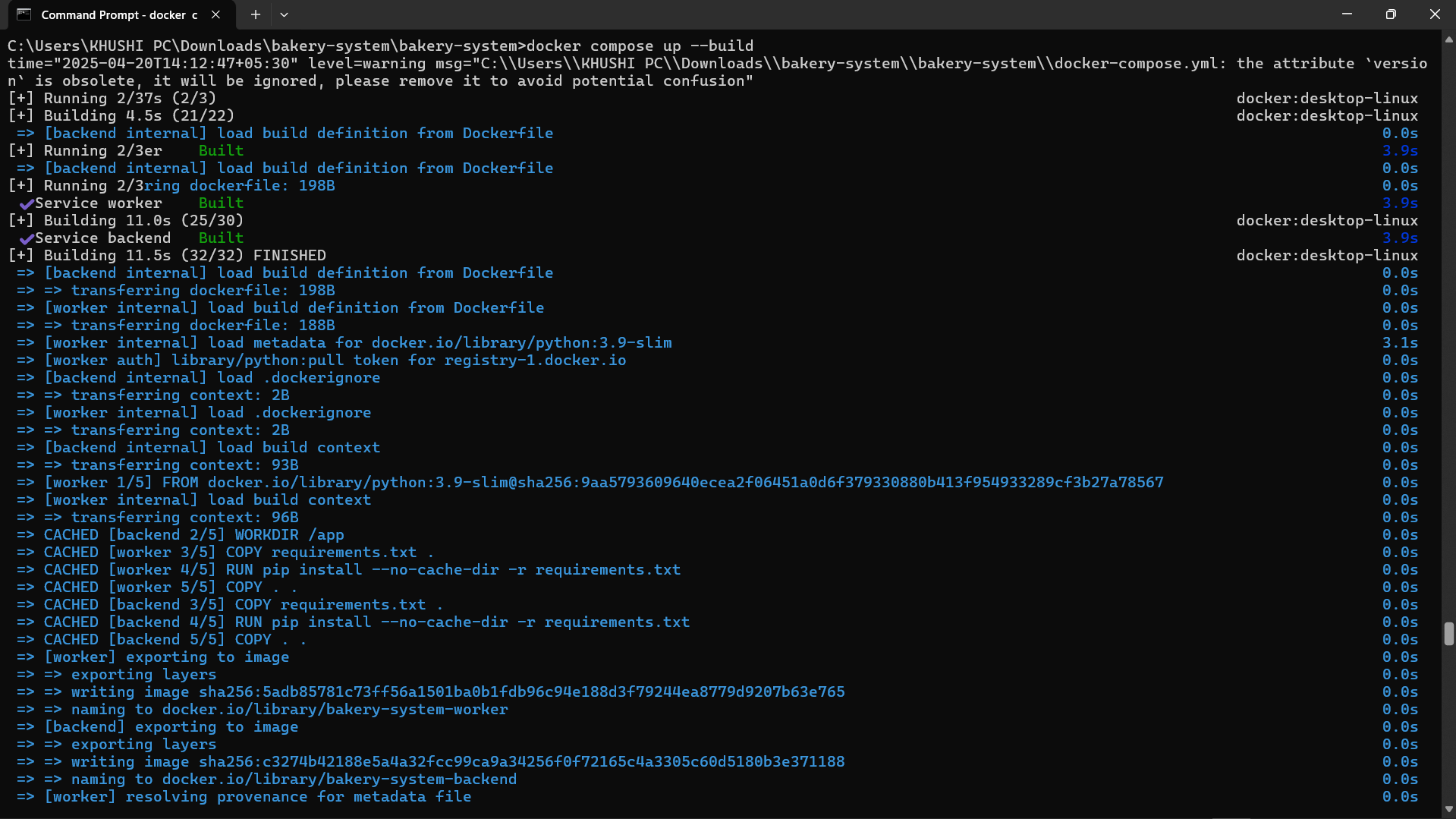
**500106017**

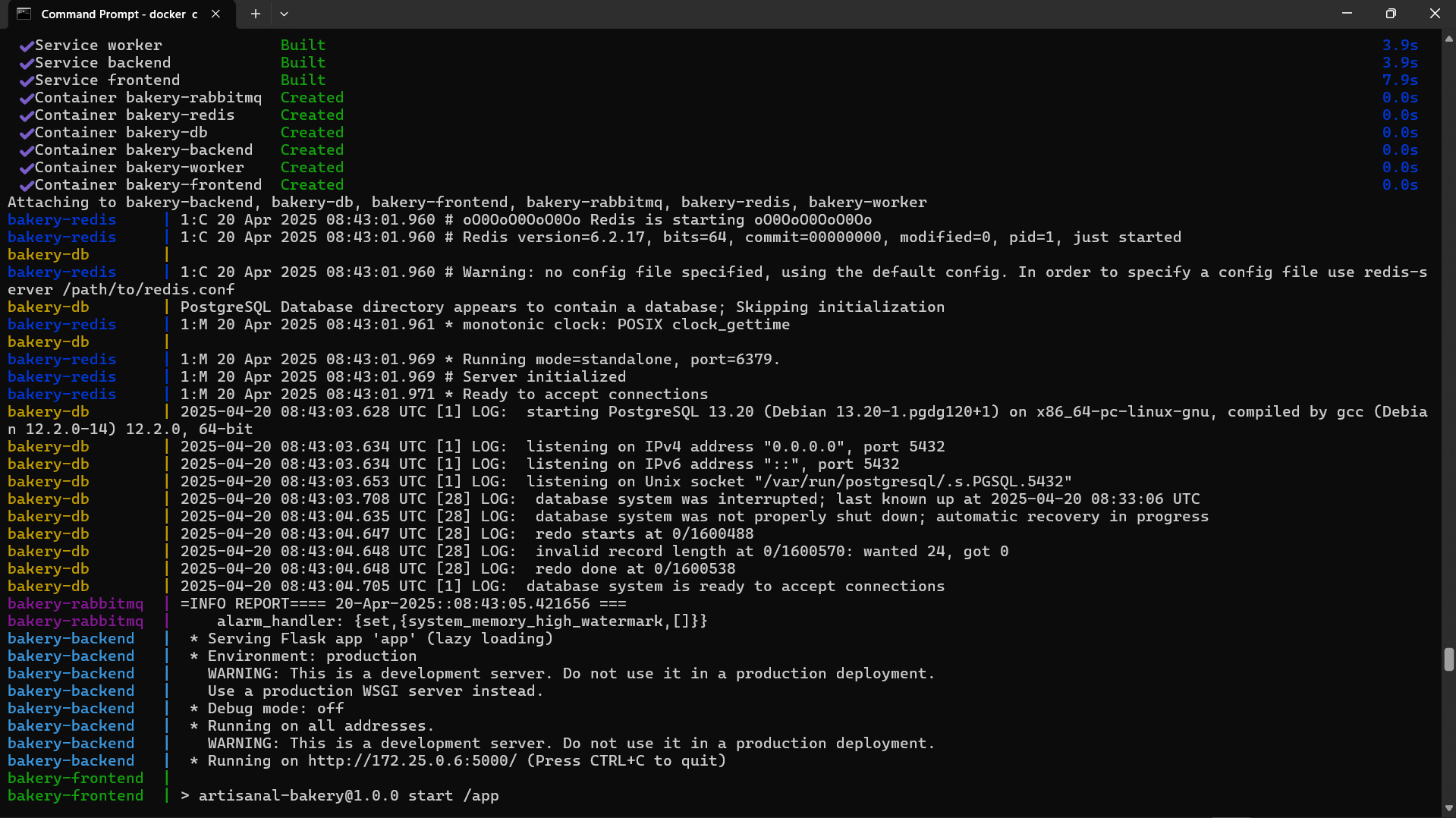
**R2142220797**

**Submitted to:**

**Dr. Srinivasan Ramachandran**







Database Container   
CREATE TABLE products (

    id SERIAL PRIMARY KEY,

    name VARCHAR(100) NOT NULL,

    description TEXT,

    price DECIMAL(10, 2) NOT NULL,

    image\_url VARCHAR(255),

    created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE orders (

    id SERIAL PRIMARY KEY,

    customer\_name VARCHAR(100) NOT NULL,

    customer\_email VARCHAR(100) NOT NULL,

    status VARCHAR(50) DEFAULT 'pending',

    created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE order\_items (

    id SERIAL PRIMARY KEY,

    order\_id INTEGER REFERENCES orders(id),

    product\_id INTEGER REFERENCES products(id),

    quantity INTEGER NOT NULL,

    unit\_price DECIMAL(10, 2) NOT NULL

);

INSERT INTO products (name, description, price, image\_url) VALUES

('Chocolate Cake', 'Rich chocolate layer cake with ganache', 29.99, 'https://images.unsplash.com/photo-1578985545062-69928b1d9587'),

('Sourdough Bread', 'Artisanal sourdough bread', 8.99, 'https://images.unsplash.com/photo-1585238342028-50d4b1a1ff6f'),

('Blueberry Muffin', 'Moist muffin loaded with blueberries', 3.99, 'https://images.unsplash.com/photo-1608198093002-ad4e0054842b'),

('Croissant', 'Buttery, flaky French pastry', 4.50, 'https://images.unsplash.com/photo-1587840171670-8b850147754e');

Volumes:

postgres data:

redis:data

rabbitmq data:

Database in compose file

services:

  postgres:

    image: postgres:13

    container\_name: bakery-db

    environment:

      POSTGRES\_USER: postgres

      POSTGRES\_PASSWORD: postgres

      POSTGRES\_DB: bakery

    volumes:

      - postgres\_data:/var/lib/postgresql/data

      - ./database/init:/docker-entrypoint-initdb.d

    ports:

      - "5432:5432"

    healthcheck:

      test: ["CMD-SHELL", "pg\_isready -U postgres"]

      interval: 10s

      timeout: 5s

      retries: 5

    restart: unless-stopped

Docker Compose file used

version: '3.8'

services:

  postgres:

    image: postgres:13

    container\_name: bakery-db

    environment:

      POSTGRES\_USER: postgres

      POSTGRES\_PASSWORD: postgres

      POSTGRES\_DB: bakery

    volumes:

      - postgres\_data:/var/lib/postgresql/data

      - ./database/init:/docker-entrypoint-initdb.d

    ports:

      - "5432:5432"

    healthcheck:

      test: ["CMD-SHELL", "pg\_isready -U postgres"]

      interval: 10s

      timeout: 5s

      retries: 5

    restart: unless-stopped

  redis:

    image: redis:6

    container\_name: bakery-redis

    ports:

      - "6379:6379"

    volumes:

      - redis\_data:/data

    healthcheck:

      test: ["CMD", "redis-cli", "ping"]

      interval: 10s

      timeout: 5s

      retries: 5

    restart: unless-stopped

  rabbitmq:

    image: rabbitmq:3-management

    container\_name: bakery-rabbitmq

    ports:

      - "5672:5672"

      - "15672:15672"

    volumes:

      - rabbitmq\_data:/var/lib/rabbitmq

    healthcheck:

      test: ["CMD", "rabbitmqctl", "status"]

      interval: 30s

      timeout: 10s

      retries: 5

    restart: unless-stopped

  backend:

    build: ./backend

    container\_name: bakery-backend

    depends\_on:

      - postgres

      - rabbitmq

      - redis

    environment:

      DB\_HOST: postgres

      DB\_NAME: bakery

      DB\_USER: postgres

      DB\_PASSWORD: postgres

      RABBITMQ\_HOST: rabbitmq

      REDIS\_HOST: redis

    ports:

      - "5000:5000"

    restart: unless-stopped

    healthcheck:

      test: ["CMD", "curl", "-f", "http://localhost:5000/health"]

      interval: 30s

      timeout: 10s

      retries: 3

  worker:

    build: ./worker

    container\_name: bakery-worker

    depends\_on:

      - postgres

      - rabbitmq

    environment:

      DB\_HOST: postgres

      DB\_NAME: bakery

      DB\_USER: postgres

      DB\_PASSWORD: postgres

      RABBITMQ\_HOST: rabbitmq

    restart: unless-stopped

  frontend:

    build: ./frontend

    container\_name: bakery-frontend

    depends\_on:

      - backend

    environment:

      REACT\_APP\_API\_URL: http://localhost:5000

    ports:

      - "3000:3000"

    restart: unless-stopped

volumes:

  postgres\_data:

  redis\_data:

  rabbitmq\_data:

Dockerfile for Frontend

FROM node:14-alpine

WORKDIR /app

# Copy package files first for better caching

COPY package\*.json ./

RUN npm install

# Copy the rest of the frontend code including public directory

COPY . .

# For React apps, consider building the project for production

# RUN npm run build

EXPOSE 3000

CMD ["npm", "start"]

**Docker file and Requirements.txt for backend**

FROM python:3.9-slim

WORKDIR /app

COPY requirements.txt .

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

COPY . .

EXPOSE 5000

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

Flask==2.0.1

Werkzeug==2.0.3 # Add this line

Flask-CORS==3.0.10

psycopg2-binary==2.9.9

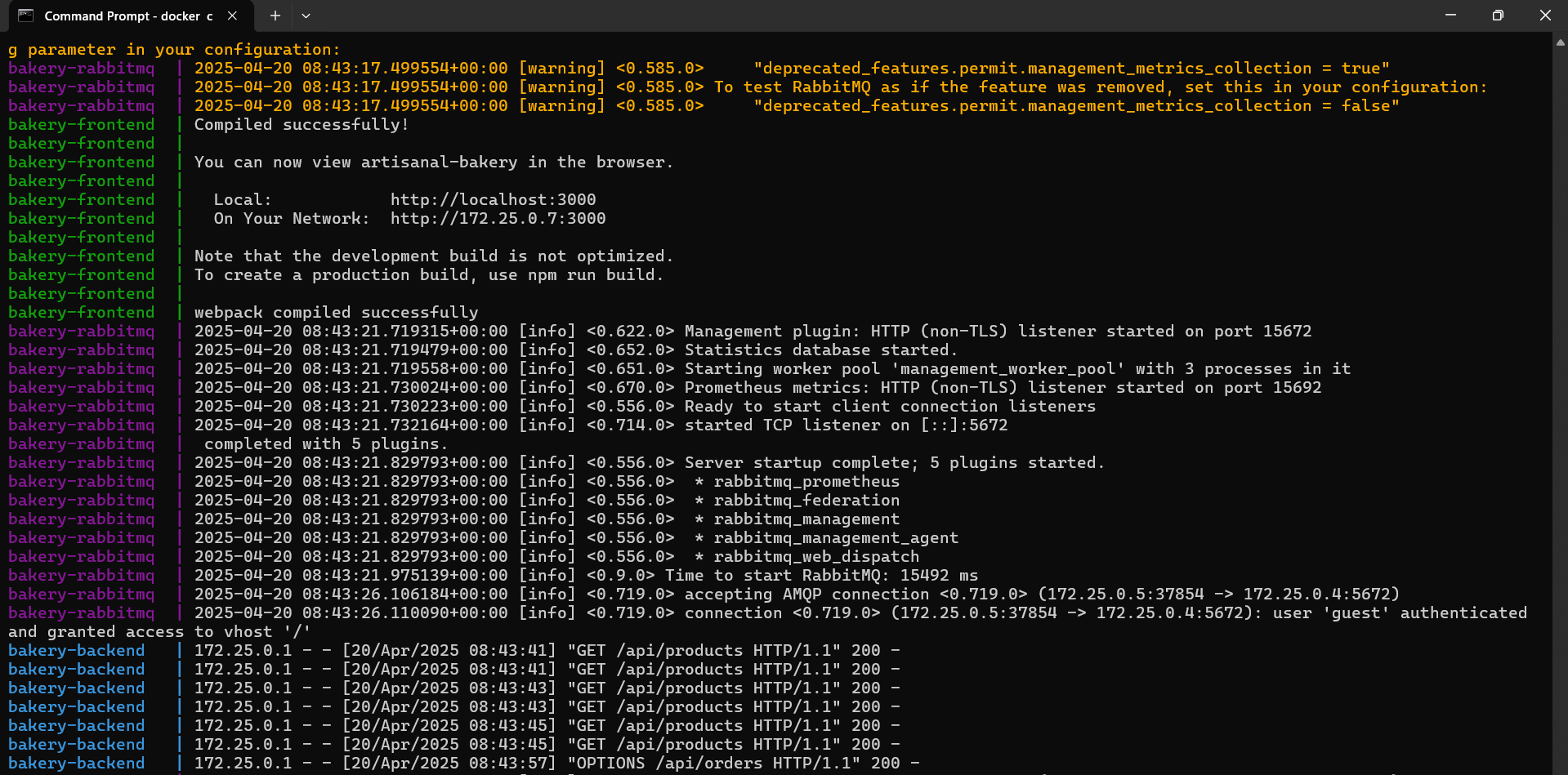
pika==1.2.0

redis==3.5.3

**Requirements.txt for worker:**  
psycopg2-binary==2.9.1

pika==1.2.0

RabbitMq



## **Additional Notes for Advanced Features**

### **1. Redis Caching**

The Redis caching implementation provides:

* Reduced database load by caching product listings
* Faster response times for frequently accessed data
* Cache invalidation when orders are placed to ensure data consistency

It has been implemented in the backend code and is imported through compose.yml

redis:

    image: redis:6

    container\_name: bakery-redis

    ports:

      - "6379:6379"

    volumes:

      - redis\_data:/data

    healthcheck:

      test: ["CMD", "redis-cli", "ping"]

      interval: 10s

      timeout: 5s

      retries: 5

    restart: unless-stopped

### **2. Worker Service with RabbitMQ**

The worker service:

* Processes orders asynchronously to improve scalability
* Updates order status in the database as processing progresses
* Implements retry logic and error handling for robustness

rabbitmq:

image: rabbitmq:3-management

container\_name: bakery-rabbitmq

ports:

- "5672:5672"

- "15672:15672"

volumes:

- rabbitmq\_data:/var/lib/rabbitmq

healthcheck:

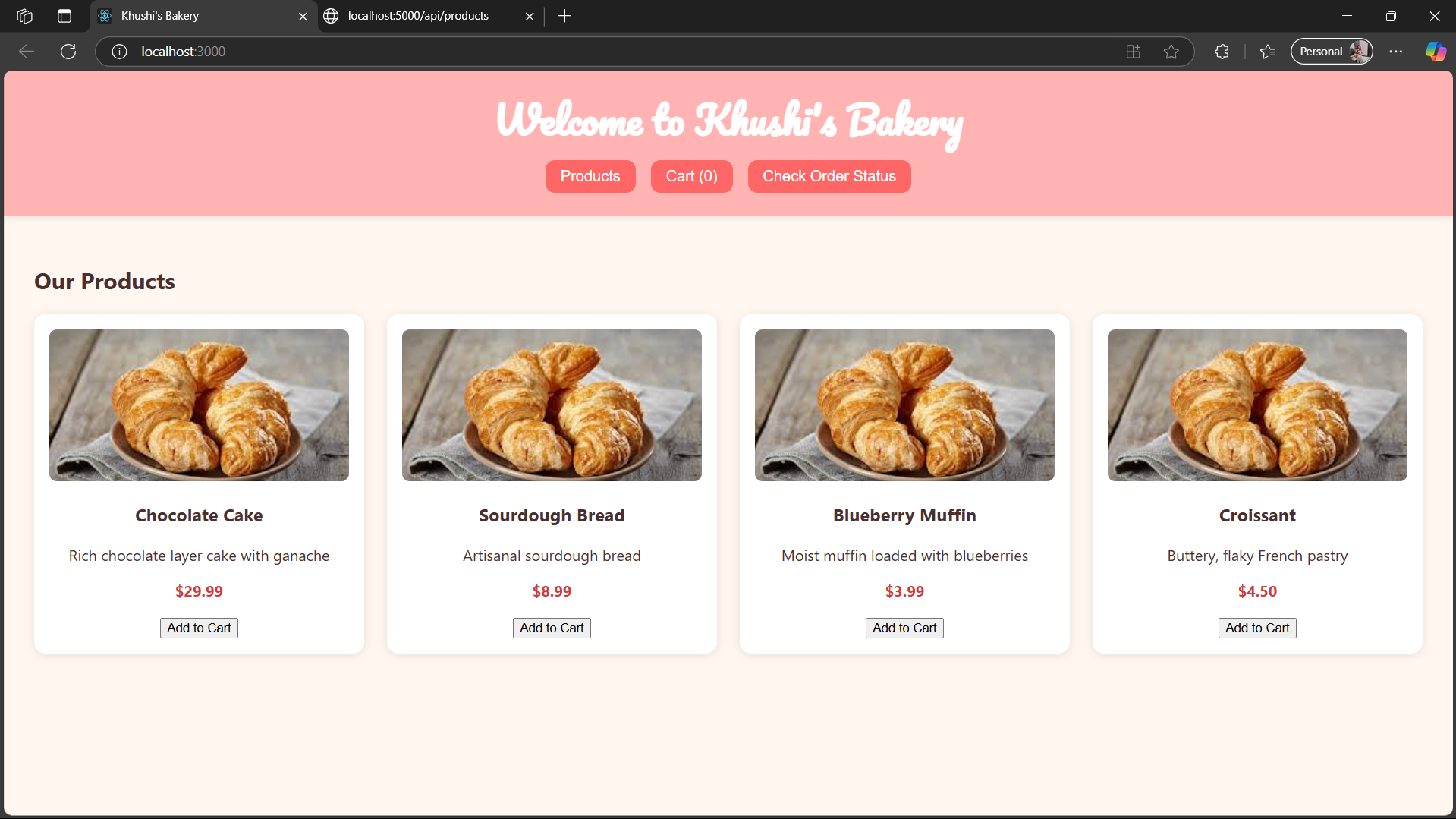
test: ["CMD", "rabbitmqctl", "status"]

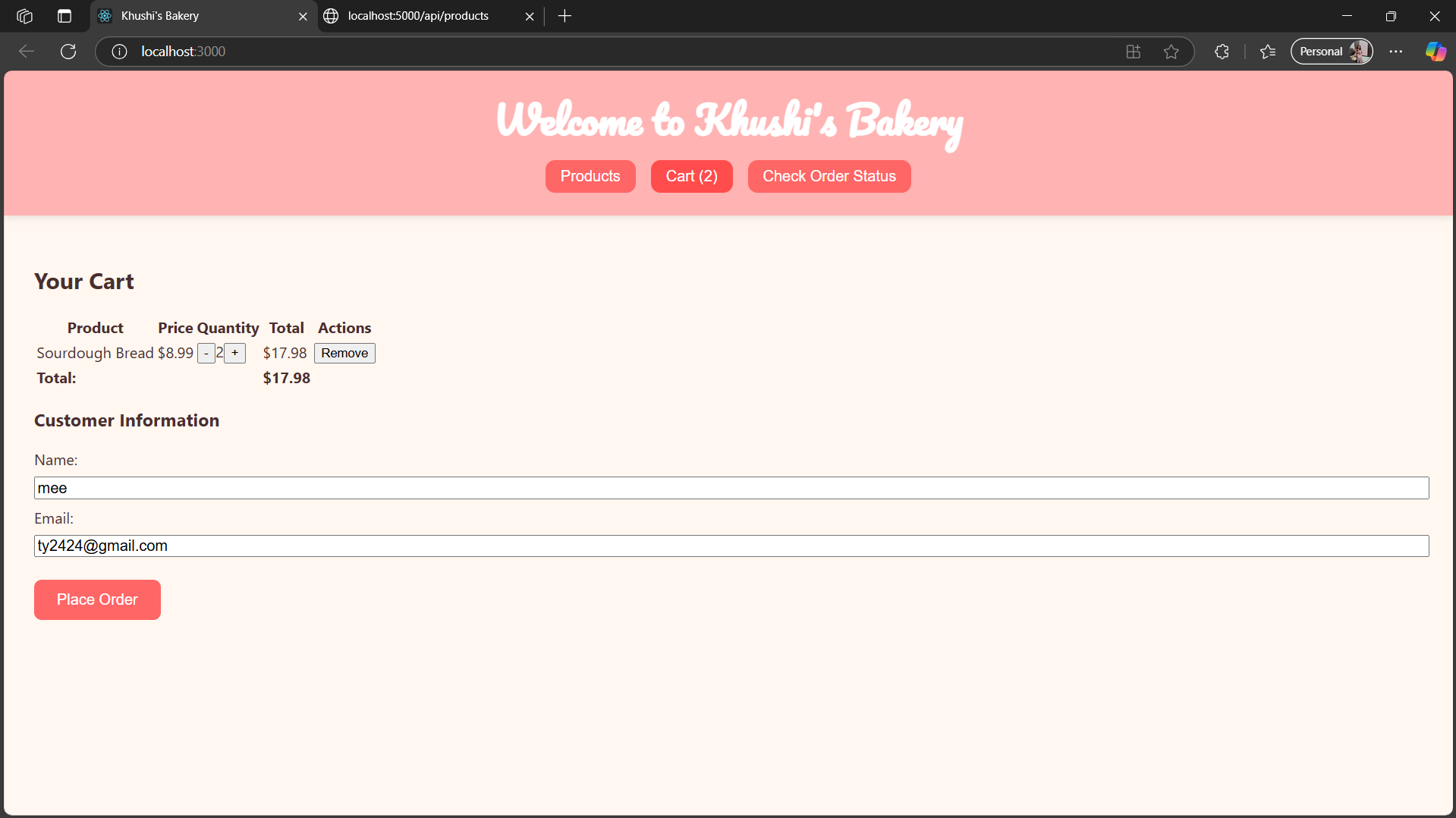
interval: 30s

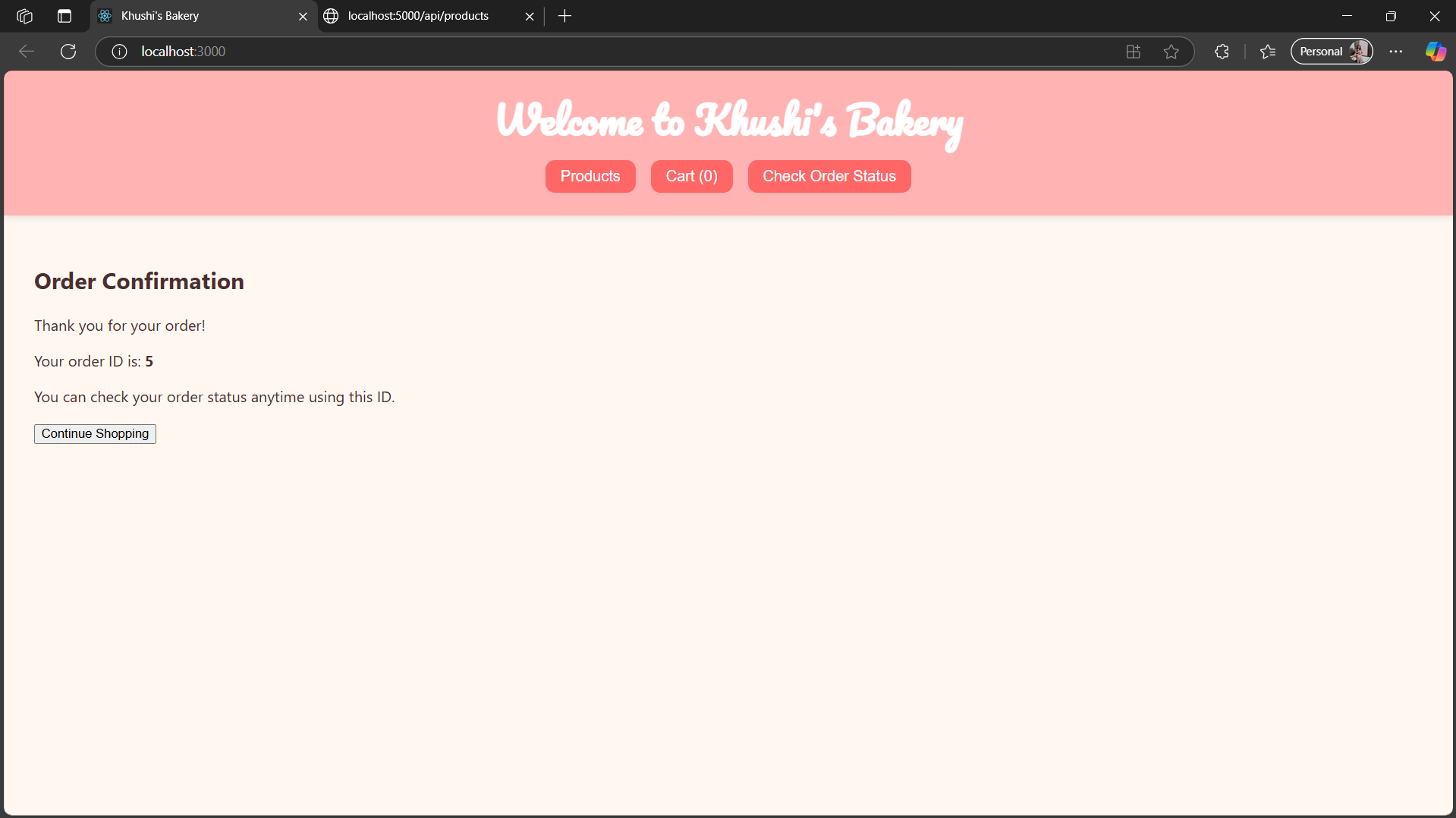
timeout: 10s

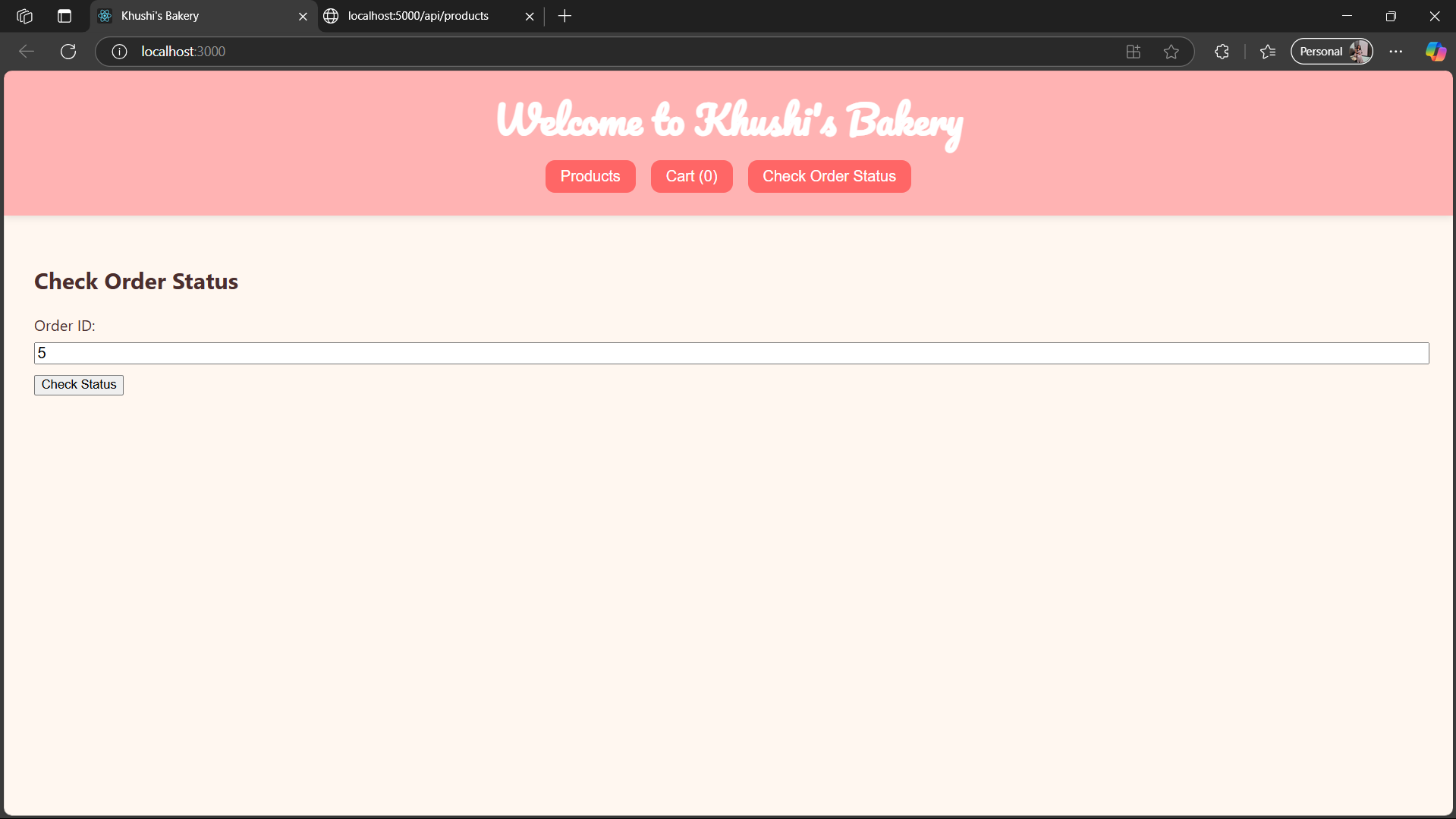
retries: 5

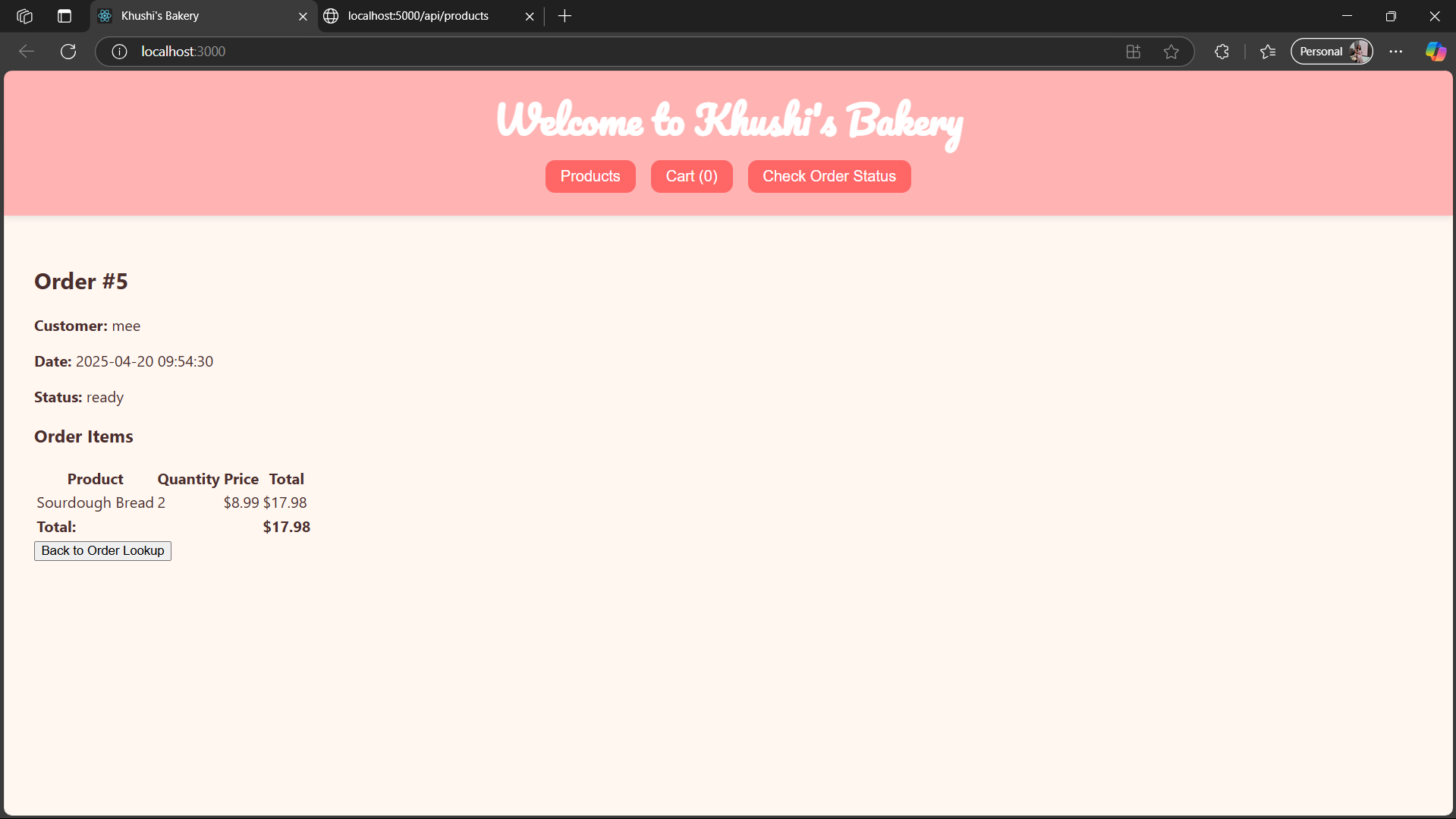
restart: unless-stopped  
  
  
**Results:**











**Google Drive Link for video:**

https://drive.google.com/drive/folders/1ZkqFlP6r2WkWxKehD-a5eVd4PLlAmIic?usp=drive\_link