

# Weather App - System Documentation

Arian Ghoochani

March 14, 2025

## Contents

<b>1</b>	<b>Overview</b>	<b>2</b>
<b>2</b>	<b>Installation Guide</b>	<b>2</b>
2.1	1. Clone the Repository . . . . .	2
2.2	2. Configure Backend Endpoint . . . . .	2
2.3	3. Build the Frontend . . . . .	2
2.4	4. Build and Run the Full-Stack Application . . . . .	2
<b>3</b>	<b>API Documentation</b>	<b>2</b>
<b>4</b>	<b>Key Configuration Files</b>	<b>3</b>
4.1	Docker Compose Configuration . . . . .	3
4.2	Nginx Configuration . . . . .	3
<b>5</b>	<b>Conclusion</b>	<b>3</b>

# 1 Overview

The **Weather App** is a full-stack web application designed to fetch and display weather data. The application consists of:

- **Frontend:** A React application that interacts with the backend API.
- **Backend:** A Flask API using Connexion for OpenAPI documentation.
- **Database:** MySQL for storing weather data.
- **Reverse Proxy:** Nginx serving the frontend and handling requests.
- **Docker:** The entire system is containerized for easy deployment.

## 2 Installation Guide

### 2.1 1. Clone the Repository

To set up the project, clone the repository:

```
git clone https://github.com/arianghoochani/Weather-App.git
cd Weather-App
```

### 2.2 2. Configure Backend Endpoint

Modify the `endpoint.json` file to point to your server:

```
{
  "weather_service": "http://<YOUR_SERVER_IP>:5000/api/weather",
  "weather_service_with_slash": "http://<YOUR_SERVER_IP>:5000/api/weather/"
}
```

### 2.3 3. Build the Frontend

```
cd weatherapp_project/weather_frontend
npm install
npm run build
```

### 2.4 4. Build and Run the Full-Stack Application

For Docker Compose v2:

```
docker compose build
docker compose up -d
```

For older versions:

```
docker-compose build
docker-compose up -d
```

## 3 API Documentation

The API is documented using OpenAPI and accessible at:

```
http://<YOUR_SERVER_IP>:5000/api/ui
```

Method	Endpoint	Description
GET	/api/weather	Retrieve all weather records
POST	/api/weather	Create a new weather record
GET	/api/weather/{city}	Retrieve weather information for a specific city
PUT	/api/weather/{city}	Update weather information for a specific city
DELETE	/api/weather/{city}	Delete weather information for a specific city

Table 1: Weather API Endpoints

## 4 Key Configuration Files

### 4.1 Docker Compose Configuration

```
services:
  flaskapp:
    build: ./weatherInfoService
    ports:
      - "5000:5000"
    depends_on:
      - mysqlldb
  mysqlldb:
    build: ./mysql
    environment:
      MYSQL_ROOT_PASSWORD: rootpassword
    ports:
      - "3306:3306"
  nginx:
    build: ./nginx
    ports:
      - "80:80"
    volumes:
      - ./weather_frontend/build:/var/www/react
```

### 4.2 Nginx Configuration

```
server {
    listen 80;
    location / {
        root /var/www/react;
    }
}
```

## 5 Conclusion

This document provides an extensive guide on setting up and deploying the Weather App. The use of **Nginx** ensures efficient static file serving and API request routing. The application is fully containerized for easy deployment.