**Activity 4: Weather Proxy API**

A professional, dynamic weather application with a NestJS backend and a React frontend. The app uses OpenWeatherMap for current conditions, debounced city suggestions, and a polished, enterprise-grade UI**.**

**Members:**

Andaya, Rhodel

Benito, Patrick Raymond P.

Cielo, Ivan Louis D.

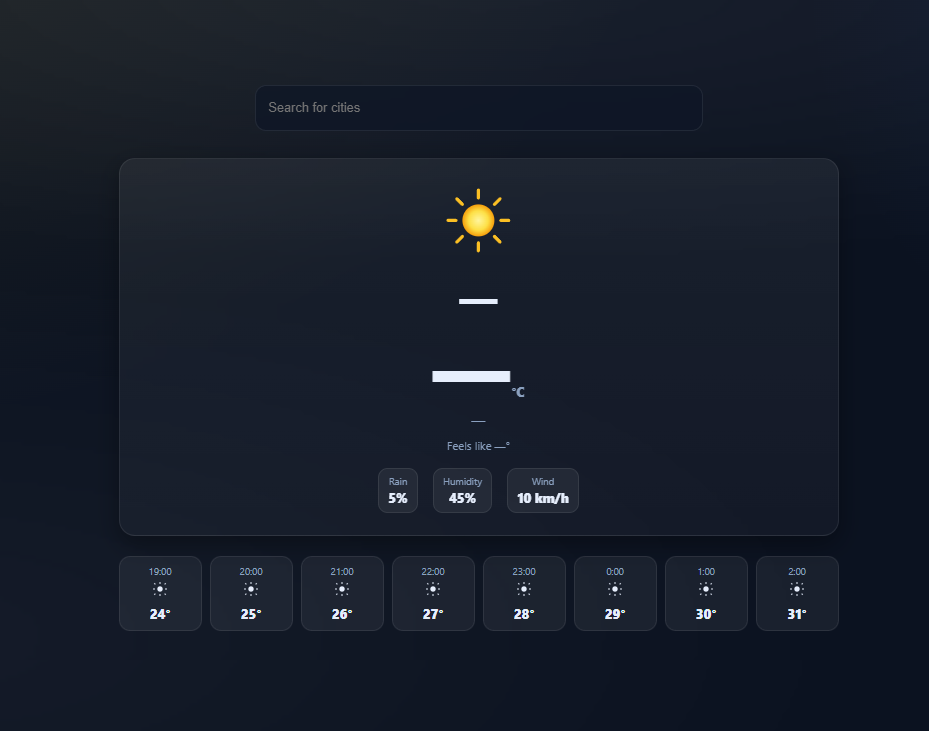
De Leon, Justin Carl V.

Dela Cruz, Althea

Dela Cruz, Filbert B.

**Steps in showing functionality:**

**Weather Dashboard - Initial State**



Upon opening the weather application, users are presented with a clean interface featuring a search bar at the top with placeholder text "Search for cities." The main display area shows a loading or default weather state with a sun icon, temperature placeholders, and "Feels like" information. Below the main weather display are metrics for Rain (5%), Humidity (45%), and Wind (10 km/h). An hourly forecast section at the bottom shows weather predictions from 19:00 to 2:00 with temperatures ranging from 24° to 31°.

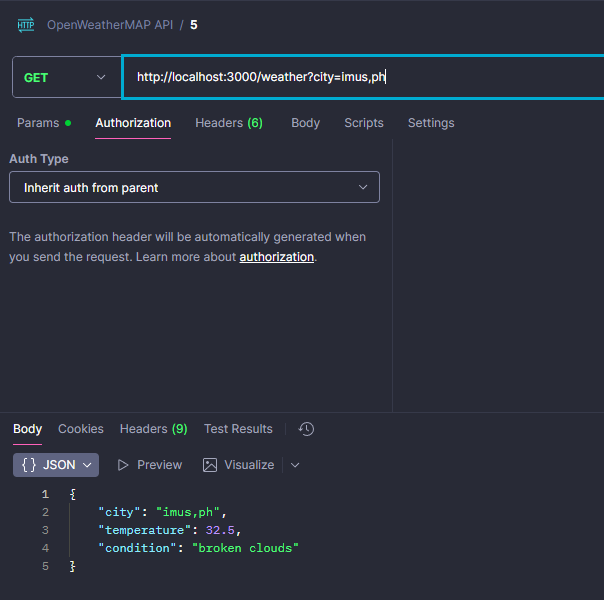
**Weather Dashboard - City Search with Global Suggestions**

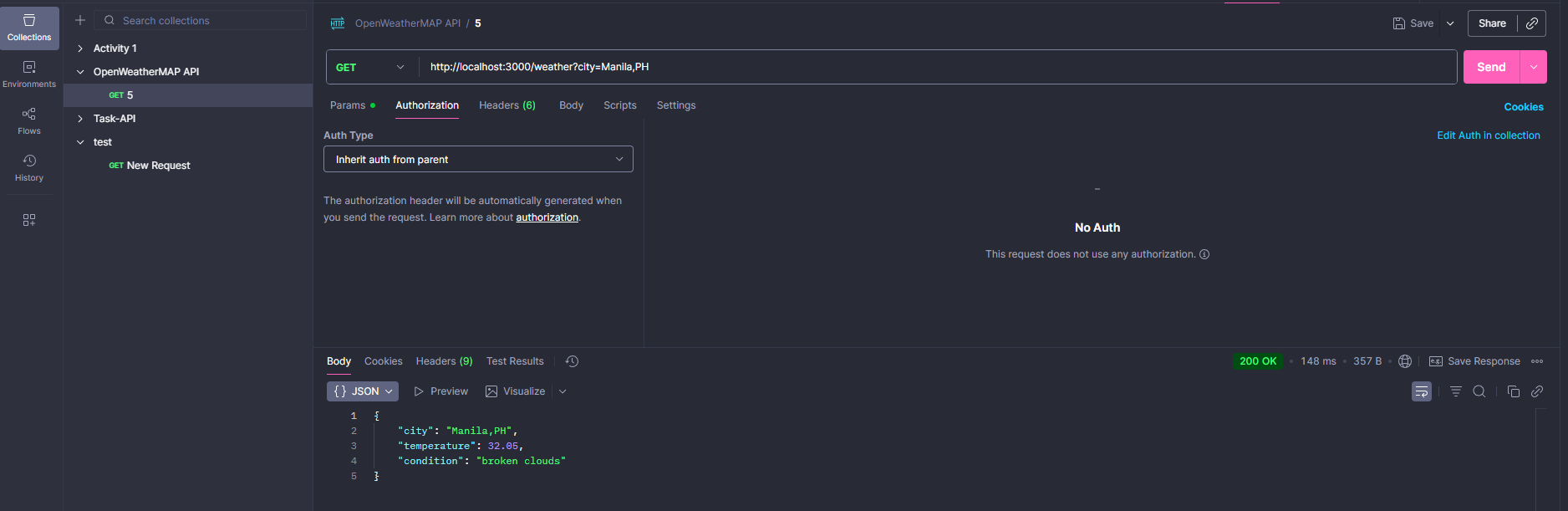




The weather application supports global city searches. When users type a city name in the search bar, a dropdown menu appears with location suggestions that include the city name, region, and country code to help users identify the exact location. For example, searching "alabama" displays multiple options like "Alabama, North West, ZA," "Alabama, Morogoro Region, TZ," and "Town of Alabama, New York, US." This prevents confusion between cities with similar names in different countries and ensures users select the correct location. Once a city is selected, the weather information updates to display current conditions, temperature, "feels like" temperature, weather metrics (Rain, Humidity, Wind), and an hourly forecast for that specific location.

**API Connectivity Testing:**





The weather application integrates with the OpenWeatherMAP API to fetch real-time weather data. Using Postman for API testing, a GET request is sent to the endpoint http://localhost:3000/weather?city=Manila,PH. The API returns weather information in JSON format, including:

* City name and country code ("Manila,PH")
* Current temperature (32.05°C)
* Weather condition description ("broken clouds")

The request receives a successful response with status code 200 OK (148 ms response time, 357 B size). This API integration enables the application to retrieve accurate, up-to-date weather information for any searched city globally.

# **Project Setup Guide**

Complete documentation for setting up and running the Weather Dashboard application.

## **Prerequisites**

* Node.js 18+ and npm
* OpenWeatherMap API Key - [Sign up here](https://openweathermap.org/api)

## **Installation**

### **Step 1: Clone and Install Dependencies**

git clone <your-repo-url>

cd activity\_4

# Backend

cd backend

npm install

# Frontend

cd ../frontend

npm install

## **Configuration**

### **Step 2: Get OpenWeatherMap API Key**

1. Go to [OpenWeatherMap API](https://openweathermap.org/api)
2. Sign up for a free account
3. Navigate to API Keys section
4. Copy your API key

### **Step 3: Set Environment Variable**

Create .env file in backend/ folder:

OPENWEATHER\_API\_KEY=your\_api\_key\_here

## **Running the Application**

### Step 4: Start Backend

cd backend

npm run start:dev

**Backend runs at http://localhost:3000**

### Step 5: Start Frontend

cd frontend

npm start

**Frontend runs at http://localhost:3000 or http://localhost:3001**

## **Troubleshooting**

### Backend Issues

Problem: "Missing OPENWEATHER\_API\_KEY"  
 Solution:

* Set the environment variable
* Restart terminal after using setx on Windows
* Check .env file exists and has correct key

Problem: "Invalid OpenWeather API key"  
 Solution:

* Verify your API key at OpenWeatherMap dashboard
* New API keys may take a few minutes to activate

Problem: Backend won't start  
 Solution: Check if port 3000 is available

### Frontend Issues

Problem: No suggestions appearing  
 Solution:

* Ensure backend is running at http://localhost:3000
* Check browser console for network errors
* Verify API key is set correctly

Problem: "City not found" errors  
 Solution:

* Use city suggestions instead of typing manually
* Check your API rate limit (60 calls/minute on free tier)
* Try using format: City, CountryCode

Problem: CORS errors  
 Solution:

* Backend must be on localhost:3000
* Frontend must be on localhost or 127.0.0.1