第7周工作汇报

下载了Sql Server 下了又删

下载了mysql 成功建立weather数据库

以下是过程

1. mysql -u root -p

存入数据：

CREATE DATABASE IF NOT EXISTS weather\_data;

USE weather\_data;

1.

CREATE TABLE IF NOT EXISTS weather (

id INT AUTO\_INCREMENT PRIMARY KEY,

date DATE NOT NULL,

weather\_type VARCHAR(50),

max\_temp INT,

min\_temp INT,

precipitation\_probability INT,

humidity\_day INT,

humidity\_night INT,

feels\_like VARCHAR(50)

);

2.

INSERT INTO weather (date, weather\_type, max\_temp, min\_temp, precipitation\_probability, humidity\_day, humidity\_night, feels\_like)

VALUES

('2025-04-14', '晴', 19, 11, 0, 30, 90, '舒适'),

('2025-04-15', '阴', 24, 14, 0, 40, 80, '温暖'),

('2025-04-16', '阴转多云', 26, 17, 0, 50, 70, '温暖'),

('2025-04-17', '晴', 30, 18, 0, 40, 60, '炎热'),

('2025-04-18', '多云转阴', 32, 18, 10, 30, 50, '酷热'),

('2025-04-19', '阴', 28, 15, 20, 60, 80, '温暖'),

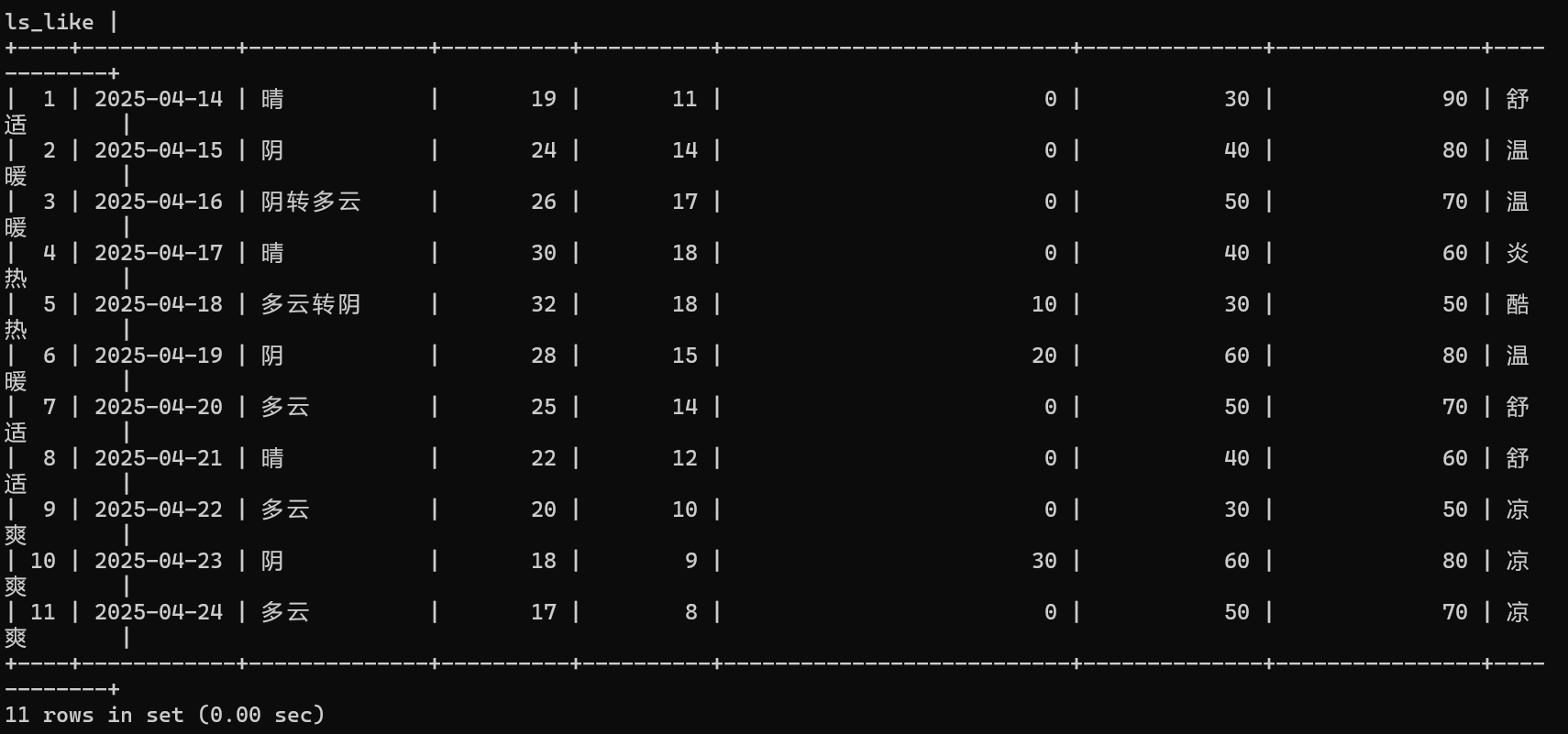
('2025-04-20', '多云', 25, 14, 0, 50, 70, '舒适'),

('2025-04-21', '晴', 22, 12, 0, 40, 60, '舒适'),

('2025-04-22', '多云', 20, 10, 0, 30, 50, '凉爽'),

('2025-04-23', '阴', 18, 9, 30, 60, 80, '凉爽'),

('2025-04-24', '多云', 17, 8, 0, 50, 70, '凉爽');



天气数据已存入weather数据库；

mkdir weather-api

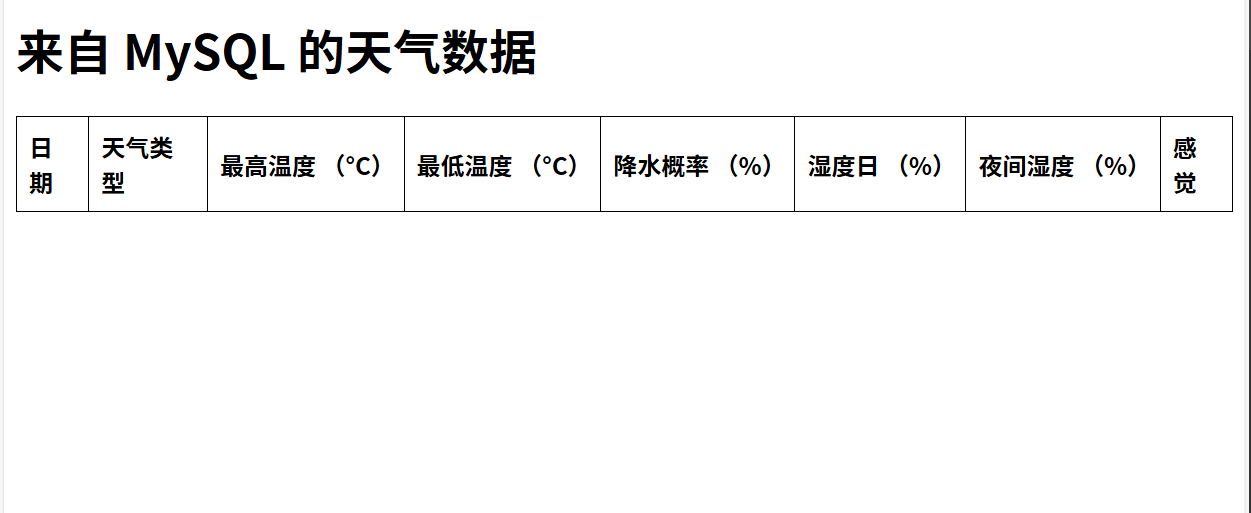
cd weather-api

npm init -y

npm install express mysql2

· 在浏览器中打开 index.html 文件，你将看到一个表格，表格中显示从 MySQL 获取的天气数据。

· · 如果一切设置正确，前端会通过 fetch 请求后端 API，后端会从数据库中获取数据并返回给前端，前端然后将这些数据展示在表格中。



创建数据库 4.15.2025

Mysql -u root -p

USE weather\_database\_name;

mysql> CREATE DATABASE weather\_db;

创建一个表

CREATE TABLE weather (

id INT AUTO\_INCREMENT PRIMARY KEY,

date DATE NOT NULL,

max\_temp DECIMAL(5,2),

min\_temp DECIMAL(5,2),

humidity DECIMAL(5,2),

feels\_like\_temp DECIMAL(5,2),

weather\_type VARCHAR(50),

precipitation\_prob DECIMAL(5,2)

);

CREATE TABLE weather (

id INT AUTO\_INCREMENT PRIMARY KEY,

date DATE NOT NULL,

max\_temp DECIMAL(5,2),

min\_temp DECIMAL(5,2),

humidity DECIMAL(5,2),

feels\_like\_temp DECIMAL(5,2),

weather\_type VARCHAR(50),

precipitation\_prob DECIMAL(5,2)

);

创建名字为weather的表格

生成weather\_data 数据

创建app.py ，将后端数据库

app.py 文件如下。

from flask import Flask, jsonify  
import mysql.connector  
  
app = Flask(\_\_name\_\_)  
  
# MySQL 配置  
db\_config = {  
 'host': 'localhost',  
 'user': 'root', # 你的 MySQL 用户名  
 'password': '123456', # 你的 MySQL 密码  
 'database': 'weather\_db' # 你的数据库名称  
}  
  
@app.route('/')  
def get\_weather():  
 # 连接到数据库  
 conn = mysql.connector.connect(\*\*db\_config)  
 cursor = conn.cursor()  
  
 # 查询天气数据  
 cursor.execute("SELECT \* FROM weather")  
 rows = cursor.fetchall()  
  
 # 返回数据作为 JSON 格式  
 weather\_data = []  
 for row in rows:  
 weather\_data.append({  
 'date': row[1],  
 'max\_temp': row[2],  
 'min\_temp': row[3],  
 'humidity': row[4],  
 'feels\_like\_temp': row[5],  
 'weather\_type': row[6],  
 'precipitation\_prob': row[7]  
 })  
  
 cursor.close()  
 conn.close()  
  
 return jsonify(weather\_data)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(debug=True)