```
/*
 * Name-Ark Nandan Singh Chauhan, 19323352
 * Express server fetching data from openweatherapi for 5 days
 */

const express = require('express');
const path = require('path');
const axios = require('axios');

//initialize express
const app = express();
// define PORT
const PORT = 3000;
app.listen(PORT, () => console.log(`Server started at Port ${PORT}`));

// let publicPath = path.resolve(__dirname, "public");
let publicPath = path.resolve(__dirname);

app.use(express.static(publicPath));

// define api url and key
let URL = "http://api.openweathermap.org/data/2.5/forecast";
let KEY = "3e2d927d4f28b456c6bc662f34350957";

app.get('/:city_name', (req, res) => {
    let city = req.params.city_name;
    let forecast = {};
    let carry_umbrella_5days = false;
    let packing_based_on_temp;
    axios.get(URL,{
        params: {
            q:city,
            APPID: KEY
        }
    })
     .then((response) => {
         //manage the response here
         api_data = response.data.list;
        //  iterate through all weather data elements
         for (el in api_data) {
            // extract date
             datetime = response.data.list[el].dt_txt;// '2020-11-07
06:00:00'
             date = datetime.substring(0,10);// '2020-11-07'
             time = datetime.substring(10,19);//  '06:00:00'
            //  Adding dates as index to forecast dictionary
            if(!forecast[date]){
                forecast[date] = {
                    isRaining: false,
                    time: [],
                    temperature: [],
                    wind_speed: [],
                    rainfall: [],
                    min_temp: [],
                    max_temp: []
                };
            }//end of if

            // add temperature, windspeed, time, min,max_temp for every 3hr
in list
```

```
            forecast[date].time.push(time);
 forecast[date].temperature.push(toCelsius(api_data[el].main.temp));
 forecast[date].min_temp.push(toCelsius(api_data[el].main.temp_min));
 forecast[date].max_temp.push(toCelsius(api_data[el].main.temp_max));
            forecast[date].wind_speed.push(api_data[el].wind.speed);

            if(api_data[el].rain){
                forecast[date].isRaining = true;
                forecast[date].rainfall.push(api_data[el].rain['3h']);
            }

            //return whether users should carry umbrella based on whether it'll rain in the upcoming 5 days
            if(forecast[date].isRaining == true){
                carry_umbrella_5days = true;
            }

         }//end of for

        //For each data point, we need to calculate averages, total rainfall and temperature range
        //temperature,average wind average
        for(key in forecast){
            forecast[key].avgTempCelsius = average(forecast[key].temperature);
            forecast[key].avgWind = average(forecast[key].wind_speed);
            // temperature range(tempRangeCelsius) has the min, max temperatures of the day
            forecast[key].tempRangeCelsius = [getMin(forecast[key].min_temp),getMax(forecast[key].max_temp)];
            forecast[key].totalRainfall = sum_list(forecast[key].rainfall);
        }

        //indicate the whether of area based on temperature
        packing_based_on_temp = temp_mapper(forecast);

        //  return the final output
         res.status(200);
         res.json({
             forecast: forecast,
             carry_umbrella_5days: carry_umbrella_5days,
             packing_based_on_temp: packing_based_on_temp
         });

     })//end of then
     .catch((error) =>{
         console.error(error);
         res.status(400);
         res.json({
             error: "This is a Bad Request!"
         })
     });
});


function average(list){
    //Returns average of the elements of the list provided
```

```
    sum=0;
    for(var i=0; i<list.length; ++i){
        sum = sum + list[i];
    }
    let avg = sum / list.length;
    return avg;
}


function toCelsius(k){
    //return celsius temperature
    return k-273.15;
}


function getMin(list){
    //get minimum of a list of elements
    let min = list[0];
    for(var i=0; i<list.length; ++i){
        if(list[i] < min){
            min = list[i];
        }
    }
    return min;
}


function getMax(list){
    //get maximum of a list of elements
    let max = 0;
    for(var i=0; i<list.length; ++i){
        if(list[i] > max){
            max = list[i];
        }
    }
    return max;
}


function sum_list(list){
    //sum of the list of elements- for rainfall measurement
    total = 0;
    if(list.length == 0){
        //if there is no rain for that day i.e list is empty
        return 0;
    }
    for(var i=0; i<list.length; ++i){
        total = total + list[i];
    }
    return total;
}


function temp_mapper(forecast){
    //return packing instruction for user based on temperature
    //finding the overall min/max temp over the period of next 5 days
    var weather_outcome;
    overall_min = forecast[key].tempRangeCelsius[0];
    overall_max = 0;
    for(key in forecast){
```

```
171
172         if(forecast[key].tempRangeCelsius[0] < overall_min){
173             overall_min = forecast[key].tempRangeCelsius[0];
174         }
175         if(forecast[key].tempRangeCelsius[1] > overall_max){
176             overall_max = forecast[key].tempRangeCelsius[1];
177         }
178     }
179     //mapping to cold, warm, hot
180     if(overall_max > 20){
181         weather_outcome = "It will be HOT over the next 5 days. Average
    Temperature > 20°C, pack light clothes."
182     }else if(overall_min>=10 && overall_max<=20){
183         weather_outcome = "It will be WARM over the next 5 days. Average
    Temperature between 10°C-20°C, pack some jackets."
184     }else{
185         weather_outcome = "It will be COLD over the next 5 days. Average
    Temperature less than 10°C, pack heavy jackets to keep warm."
186     }
187     return weather_outcome;
188 }
189
```