```
import express from "express";
import fetch from "node-fetch";
import dotenv from "dotenv";
import cors from "cors";

dotenv.config();
const app = express();
app.use(cors());
const port = 8080;

function willRain(weatherdata) {
  let arr = [];
  weatherdata.forEach((period) => {
    arr.push(period["weather"][0]["main"]);
  });
  return arr.includes("Rain");
}

function getTemps(weather) {
  // 0 K = -273.15 C
  let arr = []
  let avg = 0;
  weather.forEach((period) => {
    let temp = period["temp"]["day"] - 273.15;
    arr.push(temp)
    avg = ((avg * (arr.length-1)) + temp) / arr.length
  });

  let ret = "hot";
  if (avg < 20) ret = "warm";
  if (avg < 10 && avg > -10) ret = "cold";

  return ret;
}

app.get("/forecast", async function (req, res) {
  // ensure valid city param passed
  if (req.query["city"] == undefined || req.query["city"].length == 0) {
    res.status(400);
    res.json({ error: "invalid 'city' query params" });
    return;
  }

  let city = req.query["city"];
  let cityurl = `https://api.openweathermap.org/data/2.5/forecast?
q=${city}&appid=${process.env.appid}`;
  let response = await fetch(cityurl);
  if (response.status == 200) {
    let data = await response.json();
    let lat = data["city"]["coord"]["lat"];
    let lon = data["city"]["coord"]["lon"];

    let weatherurl = `https://api.openweathermap.org/data/2.5/onecall?
lat=${lat}&lon=${lon}&exclude=current,minutely,hourly,alerts&appid=${process.e
nv.appid}`
    response = await fetch(weatherurl);
    let weatherdata = await response.json();
    weatherdata = weatherdata["daily"]

```

```
    let airurl = `http://api.openweathermap.org/data/2.5/air_pollution?lat=${lat}&lon=${lon}&appid=${process.env.appid}`;
    response = await fetch(airurl);
    let airdata = await response.json();
    let pm2_5 = airdata["list"][0]["components"]["pm2_5"];

    res.status(200);
    res.json({
      rain: willRain(weatherdata),
      temp: getTemps(weatherdata),
      mask: pm2_5 > 10 ? true : false,
      weather: weatherdata,
    });
    return;
  }

  res.status(500);
  res.json({ error: "server error" });
  return;
});

app.listen(port, function () {
  console.log(`listening on port ${port}`);
});

```