```
// server.js for Assignment 1
// 19323238
// Rohan Taneja

const express = require("express")
const request = require("request")

const app = express()
const port = 8008

app.get("/:city", retrieveFunction)

app.listen(port, () => console.log("server.js - hosted on port " + port))

function retrieveFunction(req, res) {
  console.log("Forecast request for place: " + req.params.city)
  let city = req.params.city
  const response = request("https://api.openweathermap.org/data/2.5/forecast?q=" + city + "&units=metric" + "&APPID=19b4934ae7b535eda0af3e6fa45365dd", function(err, response, body) {
    if (err) {
      console.log("Error occurred: ", err)
    } else {
      let parameters = JSON.parse(body)
      let rainCheck = false
      let tempAvg = 0
      let summary = {
        "days": []
      }
      let resultInfo = {
        rain: false,
        tempAvg: 0,
        summary: []
      }
      let dailyInfo = {
        date: 0,
        dayTemp: 0,
        dayWind: 0,
        dayRain: 0
      }

      for (parameter in parameters.list) {
        if (parameters.list[parameter].weather[0].main == "Rain") {
          resultInfo.rainCheck = true
        }
        dailyInfo.dayTemp += parameters.list[parameter].main.temp
        dailyInfo.dayWind += parameters.list[parameter].wind.speed
        // only add rain if parameter is found
        if (parameters.list[parameter].hasOwnProperty('rain')) {
          if (parameters.list[parameter].rain.hasOwnProperty("3h")) {
            dailyInfo.dayRain += parameters.list[parameter].rain["3h"]
          }
        }
        if (parameter % 8 == 7) { // conditional block when 1 day is iterated - truncate upto 2 dec places
          //dailyInfo.day = ~~(parameter / 8) + 1 // also an alternate to this
          dailyInfo.date = (parameters.list[parameter-4].dt_txt).substring(0,10) // date from medium of the current date
          dailyInfo.dayTemp /= 8 // 3h * 8 slots = 1 day parsed from the api call - calc avg of 8 parameters
```

```
          resultInfo.tempAvg += dailyInfo.dayTemp // store for later use when goes out of loop
          dailyInfo.dayTemp = dailyInfo.dayTemp.toFixed(2), dailyInfo.dayWind = dailyInfo.dayWind.toFixed(2), dailyInfo.dayRain = dailyInfo.dayRain.toFixed(2)
          resultInfo.summary.push(dailyInfo) // add the list to summary
          dailyInfo = {
            date: 0,
            dayTemp: 0,
            dayWind: 0,
            dayRain: 0
          } // just after every 8th parameter - perform a reset
        }
      }
      resultInfo.tempAvg /= 5
      resultInfo = JSON.stringify(resultInfo)
      // console.log(resultInfo)
      res.header("Access-Control-Allow-Origin", "*") // allow access to all the clients requesting
      res.send(resultInfo)
    }
  })
}
```