BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THÀNH PHỐ HỒ CHÍ MINH

**KHOA công nghệ thông tin**



**MÔN: Phát Triển Hệ Thống Tích Hợp**

***Đề tài:***

**Ứng dụng : Sử dụng python websocket để lấy dữ liệu thời tiết về trang web cá nhân**

Lớp học phần: DHCNTT16A - 420301417501

GVHD: Trần Thị Minh Khoa

Thành phố Hồ Chí Minh, tháng 11 năm 2023

BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THÀNH PHỐ HỒ CHÍ MINH

**KHOA công nghệ thông tin**



**MÔN: Phát Triển Hệ Thống Tích Hợp**

***Đề tài:*** **Sử dụng python websocket để lấy dữ liệu thời tiết về trang web cá nhân**

Lớp học phần: DHCNTT16A - 420300207401

Tên Sinh Viên

**Đinh Hoàng Duy – 20040311**

Thành phố Hồ Chí Minh, tháng 11 năm 2023

**File server.py**

**I. Giới Thiệu**

Trong báo cáo này, chúng ta sẽ xem xét cách xây dựng một ứng dụng web đơn giản sử dụng asyncio,websocket, json và request để truy cập API thời tiết của OpenWeatherMap và hiển thị thông tin thời tiết của một thành phố cụ thể.

**II. Các Bước Thực Hiện**

**1. Import các Thư Viện và Module**

Đầu tiên, chúng ta cần import các thư viện và module cần thiết để xây dựng ứng dụng:

import asyncio

import websocket

import json

import requests

- `requests`: Gửi yêu cầu HTTP đến API của OpenWeatherMap.

- websocket: Triển khai giao tiếp 2 chiều giữa ứng dụng web và server

- Json: hỗ trợ làm việc với file Javascript

- Asyncio: Hỗ trợ lập trình bất đồng bộ

**2. Cài Đặt Thông Tin: Lấy thông tin thời tiết hiện tại, dự báo thời tiết cho 8 ngày kế tiếp và thông tin địa điểm được chọn**

api\_key = 'da6c03b7e8178e5a82ce909cf7a92078'

def get\_8\_day\_forecast(city):

url = f'http://api.openweathermap.org/data/2.5/forecast?q={city}&appid={api\_key}&units=metric'

    response = requests.get(url)

    return response.json()

def get\_weather(city):

    url = f'http://api.openweathermap.org/data/2.5/weather?q={city}&appid={api\_key}&units=metric'

    response = requests.get(url)

    return response.json()

def get\_map():

    url=f'https://maps.openweathermap.org/maps/2.0/weather/1h/TA2/4/1/6?appid={api\_key}'

    response = requests.get(url)

    return response.json()

```

- `API\_KEY`: Khóa API từ OpenWeatherMap để truy cập dữ liệu thời tiết.

**Ứng dụng Web**

**I. Mã HTML: Giao Diện Người Dùng**

**1. Mục Tiêu**

Mục tiêu của phần mã HTML này là tạo giao diện người dùng để người dùng có thể nhập tên thành phố và lấy thông tin thời tiết.

**2. Cấu Trúc Mã HTML**

<!DOCTYPE html>

<html lang="vi">

<head>

<!-- Các thuộc tính head -->

</head>

<body>

<!-- Nội dung trang web -->

</body>

</html>

**3. CSS**

/\* Định dạng giao diện người dùng \*/

body {

/\* Thêm các thuộc tính cho body \*/

}

/\* Các phần tử cụ thể \*/

.weather-container {

/\* Định dạng phần hiển thị thông tin thời tiết \*/

}

/\* Các phần tử khác \*/

/\* ... \*/

**II. Mã JavaScript: Tương Tác với API và Hiển Thị Thông Tin**

**1. Mục Tiêu**

Mục tiêu của phần mã JavaScript này là gửi yêu cầu đến server thông qua API để lấy thông tin thời tiết và hiển thị nó lên giao diện người dùng.

**2. Thời tiết hôm nay và dự báo thời tiết cho 8 ngày + Hiển thị**

const forescast =(weatherData) =>{

  const day8Data = weatherData.list;

  const day8Forecast = document.getElementById("8-day-forecast");

  day8Forecast.innerHTML = ""; // Xóa nội dung cũ

  currentday = new Date("2023-11-03");

  day8Data.forEach((dayData) => {

    const date = new Date(dayData.dt \* 1000);

    const temperature = dayData.main.temp;

    const description = dayData.weather[0].description;

    const humidity = dayData.main.humidity;

    const windspeed = dayData.wind.speed;

    const pop= dayData.pop\*100;

    const day2 = date.getDate();

    // Tạo một biến để lưu đường dẫn hình ảnh

    let weatherIconSrc = "";

    const keywordsToCheckrain = ["light rain"];

    const keywordsToCheckclouds = ["broken clouds", "overcast clouds", "scattered clouds","few clouds" ];

    const containsKeywordsrain = keywordsToCheckrain.some(keyword => description.toLowerCase().includes(keyword));

    const containsKeywordsclouds = keywordsToCheckclouds.some(keyword => description.toLowerCase().includes(keyword));

    if (containsKeywordsrain) {

      weatherIconSrc = "muanho.png";

    } else if (containsKeywordsclouds) {

      weatherIconSrc = "maytrang.png";

    }else{

      weatherIconSrc = "nangnong.png";

    }

    if (day2 != currentday.getDate()) {

      const dayCard = document.createElement("div");

      dayCard.classList.add("day-card");

      dayCard.innerHTML = `

                          <p>Thời tiết: ${description}</p>

                          <p>Nhiệt độ: ${temperature}°C</p>

                          <p>Độ ẩm: ${humidity}%</p>

                          <p>Tốc độ gió: ${windspeed}m/s</p>

                          <p>Xác xuất mưa: ${Math.floor(pop)}%</p>

                          <img src="${weatherIconSrc}" alt="${description}"> <!-- Thêm hình ảnh thời tiết -->

                      `;

      day8Forecast.appendChild(dayCard);

      currentday = date;

    }

  });

}

**3. Biểu đồ thời tiết (nhiệt độ và tốc độ gió) + Hiển thị**

const forescastchart= (weatherData)=>{

  // Lấy thẻ canvas và dữ liệu nhiệt độ và thời gian

  const day8Data = weatherData.list;

  const temperatureCanvas = document.getElementById("temperature-chart");

  const temperatureCanvas1 = document.getElementById("temperature-chart1");

  const temperatureData = [];

  const timeLabels = [];

  const windata = [];

  let currentDate1 = null;

  day8Data.forEach((dayData) => {

    const date = new Date(dayData.dt \* 1000);

    const day = date.getDate();

    const day2 = date.toLocaleDateString();

    if (currentDate1 === null || day !== currentDate1) {

      temperatureData.push(dayData.main.temp);

      windata.push(dayData.wind.speed);

      timeLabels.push(day2);

      currentDate1 = day;

    }

  });

  updateTemperatureChart(temperatureChart, temperatureCanvas, timeLabels, temperatureData,"Nhiệt độ (°C)","rgba(211, 84, 0,1.0)","rgba(255, 190, 118,1.0)")

  updateTemperatureChart(temperatureChart1, temperatureCanvas1, timeLabels, windata,"Tốc độ gió(m/s)","rgba(9, 132, 227,1.0)","rgba(129, 236, 236,1.0)")

}

const updateTemperatureChart=(temperatureChart, temperatureCanvas, timeLabels, temperatureData,labels,color1,color2)=> {

  if (temperatureChart) {

    temperatureChart.data.labels = timeLabels;

    temperatureChart.data.datasets[0].data = temperatureData;

    temperatureChart.update();

  } else {

    temperatureChart = new Chart(temperatureCanvas, {

      type: "line",

      data: {

        labels: timeLabels,

        datasets: [

          {

            label: labels,

            data: temperatureData,

            borderColor: color1,

            borderWidth: 2,

            tension: 0.4,

            fill: true,

            backgroundColor: color2,

          },

        ],

      },

      options: {

        scales: {

          x: {

            position: 'top',

            title: {

              display: true,

            },

          },

          y: {

            title: {

              display: true,

              text: labels,

            },

          },

        },

      },

    });

  }

}