**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NÔNG LÂM TPHCM**

**KHOA CÔNG NGHỆ THÔNG TIN**



**Môn: Data Warehouse**

**GVHD: Th.S Nguyễn Đức Công Song**

**Đề tài: Hệ Thống Báo Giá Xe Đạp**

**Sinh viên thực hiện:**

21130558 – Mai Xuân Thức

20130238 – Hoàng Nhựt Duy

21130340 – Lê Ngọc Hà

21130448 – Cao Thành Nam

21130168 – Võ Vũ Minh Tâm

**Mục lục**

[**Lời mở đầu** 3](#_Toc182641043)

[**LỜI CÁM ƠN** 4](#_Toc182641044)

[**NHẬN XÉT** 5](#_Toc182641045)

[**Revision History** 6](#_Toc182641046)

[**I.** **KIẾN TRÚC HỆ THỐNG** 6](#_Toc182641047)

[**II.** **MÔ TẢ DỮ LIỆU ĐẦU VÀO (DATAFEED SPECIFICATION)** 7](#_Toc182641048)

[**1.** **MÔ TẢ DỮ LIỆU ĐẦU VÀO** 7](#_Toc182641049)

[**III.** **CẤU TRÚC CSDL CONTROL** 7](#_Toc182641050)

[**1.** **CẤU TRÚC CƠ BẢN, KIỂU DỮ LIỆU** 7](#_Toc182641051)

[**a.** **Config** 7](#_Toc182641052)

[**b.** **logs** 7](#_Toc182641053)

[**c.** **Status** 8](#_Toc182641054)

[**d.** **dateDim** 8](#_Toc182641055)

[**e.** **monthDim** 9](#_Toc182641056)

[**2.** **CLASS DIAGRAM** 9](#_Toc182641057)

[**IV.** **CẤU TRÚC CSDL STAGING** 10](#_Toc182641058)

[**1.** **CẤU TRÚC CƠ BẢN, KIỂU DỮ LIỆU** 10](#_Toc182641059)

[**a.** **bikes** 10](#_Toc182641060)

[**2.** **CLASS DIAGRAM** 11](#_Toc182641061)

[**V.** **CẤU TRÚC CSDL CỦA DATAWAREHOUSE** 11](#_Toc182641062)

[**1.** **CẤU TRÚC CƠ BẢN, KIỂU DỮ LIỆU** 11](#_Toc182641063)

[**f.** **Brand** 11](#_Toc182641064)

[**g.** **Bicycle** 11](#_Toc182641065)

[**h.** **Category** 12](#_Toc182641066)

[**i.** **BicycleDetails** 12](#_Toc182641067)

[**2.** **CLASS DIAGRAM** 12](#_Toc182641068)

[**VI.** **CHỨC NĂNG TỰ ĐỘNG** 13](#_Toc182641069)

[**1.** **Server** 13](#_Toc182641070)

[**1.1.** **ConfigController** 13](#_Toc182641071)

[**1.2.** **DateDimController** 14](#_Toc182641072)

[**1.3.** **EmailController** 15](#_Toc182641073)

[**1.4.** **LogController** 15](#_Toc182641074)

[**1.5.** **StatusController** 16](#_Toc182641075)

[**1.6.** **BikeController** 17](#_Toc182641076)

[**2.** **Client** 18](#_Toc182641077)

[**2.1.** **Config chuẩn bị để lấy dữ liệu** 18](#_Toc182641078)

[**2.2.** **Lấy dữ liệu từ trang web bike2school.vn** 18](#_Toc182641079)

[**2.3.** **Lấy dữ liệu từ trang web xedapgiakho.com** 20](#_Toc182641080)

[**VII.** **CÀI ĐẶT ĐỂ LẤY DỮ LIỆU TỰ ĐỘNG** 21](#_Toc182641081)

[**2.** **Cài đặt** 21](#_Toc182641082)

[**VIII.** **HƯỚNG DẪN CHỨC NĂNG CHẠY BẰNG TAY** 27](#_Toc182641083)

[**IX.** **SƠ ĐỒ LUỒNG THỰC HIỆN VIỆC LẤY DỮ LIỆU VÀO HỆ THỐNG** 28](#_Toc182641084)

# **Lời mở đầu**

Trong thời đại công nghệ số ngày nay, việc quản lý và phân tích dữ liệu đóng vai trò quan trọng trong việc ra quyết định chiến lược cho các doanh nghiệp. Đặc biệt trong ngành bán lẻ, dữ liệu về sản phẩm, giá cả, và hành vi người tiêu dùng trở thành nguồn tài nguyên quý giá giúp nâng cao hiệu quả kinh doanh. Đề tài “Báo cáo giá xe đạp” ra đời nhằm giải quyết nhu cầu lưu trữ, quản lý, và phân tích dữ liệu giá cả từ các nguồn khác nhau, hỗ trợ doanh nghiệp trong việc cung cấp thông tin chính xác và kịp thời. Kho dữ liệu (Data Warehouse) không chỉ giúp lưu trữ dữ liệu từ nhiều nguồn mà còn cung cấp các công cụ phân tích mạnh mẽ để dự đoán xu hướng giá, quản lý hàng tồn kho, và tối ưu hóa chiến lược kinh doanh. Bằng cách tổng hợp dữ liệu giá từ các website bán xe đạp, hệ thống sẽ cho phép doanh nghiệp theo dõi biến động giá cả, so sánh giữa các thương hiệu, và phát hiện những cơ hội kinh doanh tiềm năng. Báo cáo này sẽ trình bày quá trình xây dựng hệ thống Data Warehouse cho dữ liệu giá xe đạp, bao gồm các bước thu thập dữ liệu, thiết kế hệ thống kho dữ liệu, và ứng dụng phân tích nhằm hỗ trợ doanh nghiệp ra quyết định dựa trên dữ liệu. Thông qua nghiên cứu này, chúng tôi kỳ vọng sẽ giúp các doanh nghiệp bán lẻ xe đạp không chỉ nâng cao năng lực quản lý dữ liệu mà còn phát triển chiến lược cạnh tranh hiệu quả trong thị trường.

**LỜI CÁM ƠN**

Chúng em xin chân thành cảm ơn Thầy đã tận tình hướng dẫn và truyền đạt những kiến thức quý báu trong suốt khóa học Data Warehouse. Nhờ sự chỉ dẫn tận tâm của Thầy, chúng em đã có cơ hội tiếp cận với những khái niệm và kỹ thuật mới mẻ, cũng như thực hành và phát triển các dự án thực tế. Khóa học này đã trang bị cho chúng em không chỉ kiến thức chuyên môn mà còn cả những kỹ năng quan trọng trong công việc và cuộc sống. Những bài học và kinh nghiệm quý báu mà chúng em đã nhận được sẽ là hành trang vững chắc trên con đường phát triển sự nghiệp trong lĩnh vực công nghệ thông tin. Song cũng không tránh khỏi những thiếu sót, mong thầy sẽ tận tình chỉ bảo thêm cho chúng em.

**NHẬN XÉT**

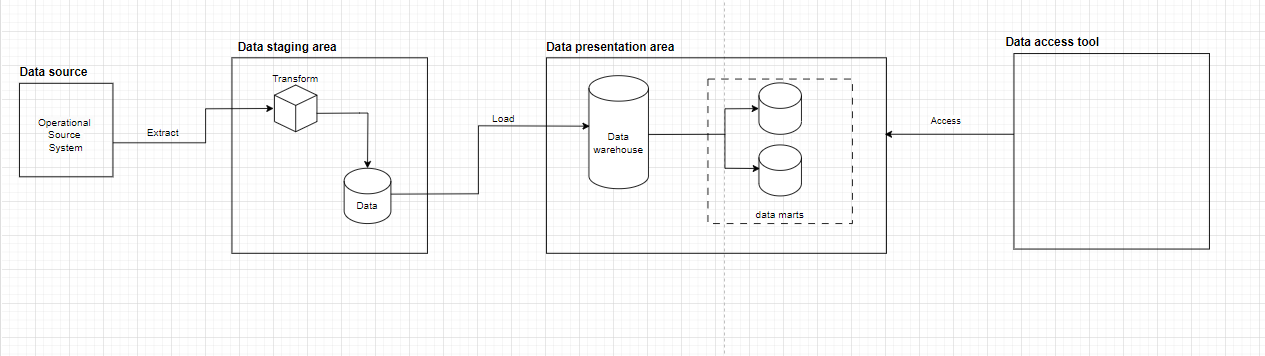
Ngày 02 tháng 06 năm 2024

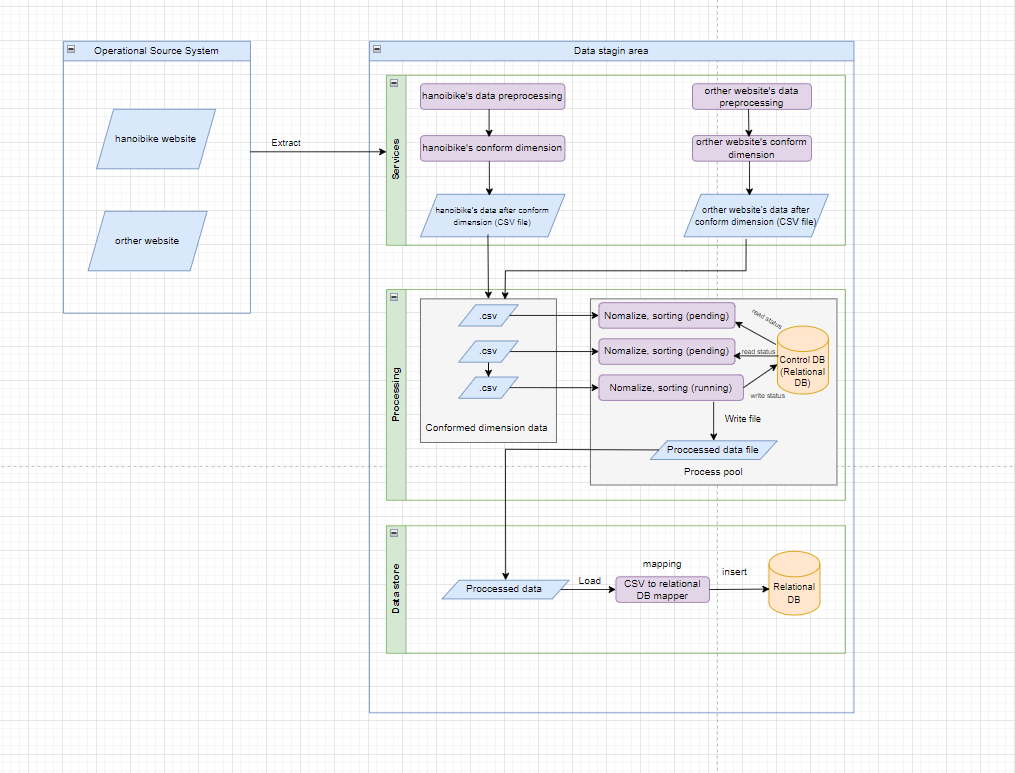
(**Ký tên**)

**Revision History**

|  |  |  |
| --- | --- | --- |
| **Ngày** | **Tác giả** | **Nội dung** |
| 13/11/2024 | C.T.Nam | Sửa lại data feed, thêm mô tả cho CSDL staging và control |
| 15/11/2024 | C.T.Nam | Thêm Mô tả chức năng, cài đặt ứng dụng chạy tự động |
| 16/11/2024 | C.T.Nam | Thêm hướng dẫn cách chạy bằng tay |

# **KIẾN TRÚC HỆ THỐNG**





# **MÔ TẢ DỮ LIỆU ĐẦU VÀO (DATAFEED SPECIFICATION)**

## **MÔ TẢ DỮ LIỆU ĐẦU VÀO**

|  |  |
| --- | --- |
| **Trường** | **Kiểu dữ liệu** |
| id | String |
| name | String |
| price | String |
| price\_sale | String |
| brand | String |
| color | String |
| size | String |
| status | String |
| description\_part1 | String |
| description\_part2 | String |
| description\_part3 | String |
| timeStartScrape | LocalDateTime |
| timeEndScrape | LocalDateTime |
| timeStartInsert | LocalDateTime |

# **CẤU TRÚC CSDL CONTROL**

## **CẤU TRÚC CƠ BẢN, KIỂU DỮ LIỆU**

### **Config**

|  |  |  |
| --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Mô tả** |
| id | INT(6) PRIMARY KEY AUTO\_INCREMENT | Định danh duy nhất cho từng website. |
| website | VARCHAR(255) | Tên của trang web. |
| saveFolder | VARCHAR(255) | Đường dẫn đến thư mục lưu trữ sau khi lấy dữ liệu. |
| keyword | VARCHAR(255) | Từ khóa của trang web để hỗ trợ quá trình tìm kiếm dữ liệu. |
| scrapeTimes | INT(6) | Số lần đã lấy dữ liệu từ trang web. |

### **logs**

|  |  |  |
| --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Mô tả** |
| id | INT(6) PRIMARY KEY AUTO\_INCREMENT | Định danh duy nhất cho từng log. |
| message | VARCHAR(255) | Ghi chú hoặc mô tả cho log. |
| quantity | INT(6) | Số lượng dữ liệu đã lấy. |
| statusId | INT(6) | Trạng thái của log (có thể tham chiếu đến bảng trạng thái nếu cần). |
| websiteId | INT(6) | ID của website, liên kết với bảng "websites". |

|  |  |  |
| --- | --- | --- |
| timeStart | DATETIME | Thời gian bắt đầu truy cập vào trang web. |
| timeEnd | DATETIME | Thời gian kết thúc truy cập vào trang web. |

### **Status**

|  |  |  |
| --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Mô tả** |
| id | INT(6) PRIMARY KEY AUTO\_INCREMENT | Định danh duy nhất cho từng trạng thái. |
| name | VARCHAR(255) | Tên của trạng thái (ví dụ: active, inactive).  **RUNNING**: hệ thống đang trong quá trình lấy dữ liệu  **WAITING**: thể hiện trong ngày hôm nay website này chưa được lấy dữ liệu và đang chờ để lấy dữ liệu  **FAILED**: thể hiện nếu số lượng dòng lấy dữ liệu bằng không  **COMPLETE**: đã hoàn thành lấy dữ liệu thành công |

### **dateDim**

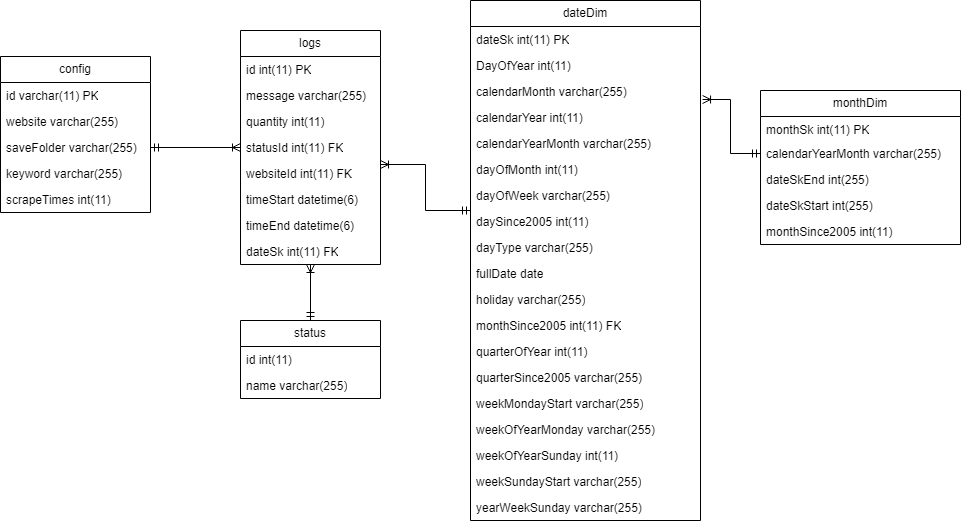
|  |  |  |
| --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Mô tả** |
| dateSk | INT(6) PRIMARY KEY | Khóa chính, duy nhất định danh từng ngày. |
| DayOfYear | INT(6) | Ngày thứ i trong năm (1, 2, 3,...). |
| calendarMonth | VARCHAR(255) | Tháng được lưu trữ ở dạng chữ (January, February,...). |
| calendarYear | YEAR | Năm (2005, 2006, 2007,...). |
| calendarYearMonth | VARCHAR(255) | Năm và tháng (2005-Jan, 2005-Feb,...). |
| dayOfMonth | INT(6) | Ngày thứ i trong tháng (1, 2, 3,...). |
| dayOfWeek | VARCHAR(255) | Thứ trong tuần lưu ở dạng chữ (Monday, Tuesday,...). |
| daySince2005 | INT(6) | Ngày thứ i tính từ ngày 01/01/2005. |
| dayType | VARCHAR(255) | Ngày trong tuần hay cuối tuần (weekday, weekend). |
| fullDate | DATE | Ngày tháng năm đầy đủ (YYYY/mm/dd). |
| holiday | VARCHAR(255) | Ngày lễ hay không (non-holiday, holiday). |
| monthSince2005 | INT(6) | Tháng thứ i tính từ ngày 01/01/2005. |
| quarterOfYear | INT(6) | Quý của năm (1, 2, 3, 4). |

|  |  |  |
| --- | --- | --- |
| quarterSince2005 | VARCHAR(255) | Quý của năm (ví dụ: 2005-Q1, 2005-Q2). |
| weekMondayStart | DATE | Ngày thứ 2 của tuần (định dạng dd/mm/YYYY). |
| weekOfYearMonday | INT(6) | Tuần thứ i(1…53) của năm, đầu tuần tính từ thứ hai. |
| weekOfYearSunday | INT(6) | Tuần thứ i(1...52) của năm, đầu tuần tính từ Chủ nhật. |
| weekSundayStart | DATE | Ngày Chủ nhật của tuần (định dạng dd/mm/YYYY). |
| yearWeekSunday | VARCHAR(255) | Năm và tuần, đầu tuần tính từ Chủ nhật (2005-W01, 2005-W02). |

### **monthDim**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Trường** | | |  | | --- | | **Trường** | | |  | | --- | | **Trường** | |
| monthSk | INT(6) PRIMARY KEY | Khóa ngoại, liên kết với cột monthSince2005 trong dateDim. |
| calendarYearMonth | VARCHAR(255) | Năm và tháng (2005-Jan, 2005-Feb,...). |
| dateSkEnd | INT(6) | Khóa ngoại liên kết đến cột dateSk trong dateDim, thể hiện ngày kết thúc của tháng. |
| dateSkStart | INT(6) | Khóa ngoại liên kết đến cột dateSk trong dateDim, thể hiện ngày bắt đầu của tháng. |
| monthSince2005 | INT(6) | Tháng thứ i(1, 2, 3, 4…) tính từ ngày 01/01/2005. |

1. **CLASS DIAGRAM**

****

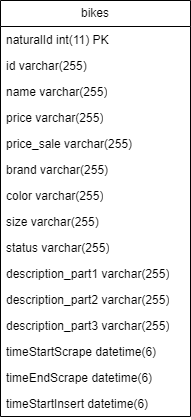
1. **CẤU TRÚC CSDL STAGING**

## **CẤU TRÚC CƠ BẢN, KIỂU DỮ LIỆU**

### **bikes**

|  |  |  |
| --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Mô tả** |
| naturalId | INT(11) PRIMARY KEY | Định danh duy nhất cho từng xe đạp (có thể là mã tự nhiên hoặc định danh ngoài). |
| id | VARCHAR(255) | Mã xe đạp (có thể là SKU hoặc mã do hệ thống quản lý). |
| name | VARCHAR(255) | Tên của xe đạp. |
| price | VARCHAR(255) | Giá bán của xe đạp. |
| price\_sale | VARCHAR(255) | Giá bán đã giảm (nếu có giảm giá). |
| brand | VARCHAR(255) | Thương hiệu của xe đạp. |
| color | VARCHAR(255) | Màu sắc của xe đạp. |
| size | VARCHAR(255) | Kích thước xe đạp. |
| status | VARCHAR(255) | Tình trạng xe đạp. |
| description\_part1 | VARCHAR(255) | Các thông số kỹ thuật (phần 1). |
| description\_part2 | VARCHAR(255) | Các thông số kỹ thuật (phần 2). |
| description\_part3 | VARCHAR(255) | Các thông số kỹ thuật (phần 3). |
| timeStartScrape | DATETIME | Thời gian bắt đầu lấy dữ liệu từ nguồn. |
| timeEndScrape | DATETIME | Thời gian kết thúc lấy dữ liệu từ nguồn. |
| timeStartInsert | DATETIME | Thời gian thêm vào cơ sở dữ liệu. |

1. **CLASS DIAGRAM**

****

1. **CẤU TRÚC CSDL CỦA DATAWAREHOUSE**

## **CẤU TRÚC CƠ BẢN, KIỂU DỮ LIỆU**

### **Brand**

|  |  |  |
| --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Mô tả** |
| BrandID | INT(6) (PK) | Khóa chính |
| BrandName | VARCHAR | Tên thương hiệu |

### **Bicycle**

|  |  |  |
| --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Mô tả** |
| BicycleID | INT(6) (PK) | Khóa chính |
| Name | VARCHAR | Tên sản phẩm |
| BrandID | INT(6) (FK) | Liên kết với thương hiệu |
| Price | DECIMAL | Giá |
| CategoryID | INT(6) (FK) | Liên kết với danh mục sản phẩm |
| Image | VARCHAR | Đường dẫn hình ảnh sản phẩm |
| Description | TEXT | Mô tả chi tiết sản phẩm |
| Price\_sale | DECIMAL | Giá đã giảm |

|  |  |  |
| --- | --- | --- |
| Size | INT(6) | Kích thước |
| Status | VARCHAR | Tình trạng |

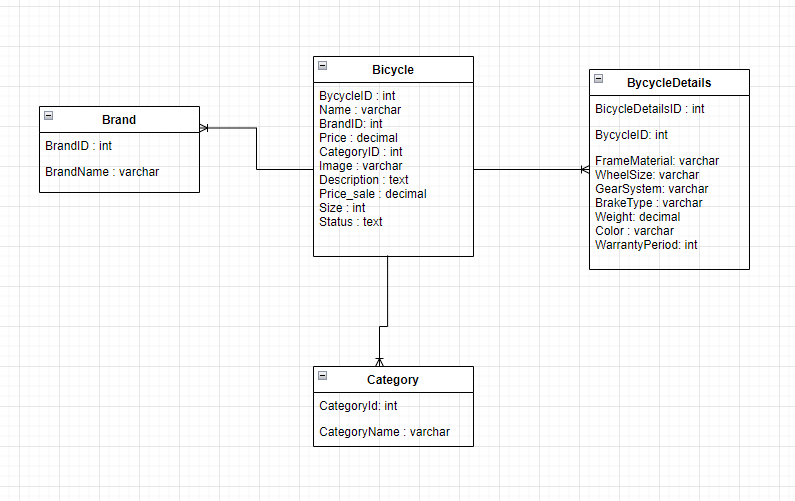
### **Category**

|  |  |  |
| --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Mô tả** |
| CategoryID | INT(6) (PK) | |  | | --- | |  |  |  | | --- | | Khóa chính | |
| CategoryName | VARCHAR | |  | | --- | |  |  |  | | --- | | Tên danh mục | |

### **BicycleDetails**

|  |  |  |
| --- | --- | --- |
| **Trường** | **Kiểu dữ liệu** | **Mô tả** |
| BicycleDetailsID | INT(6)(PK) | |  | | --- | | Khóa chính | |
| BicycleID | INT(6)(FK) | Liên kết với bicycle |
| FrameMaterial | VARCHAR | Chất liệu khung xe |
| WheelSize | VARCHAR | Kích cỡ bánh xe |
| GearSystem | VARCHAR | Hệ thống bánh răng |
| BrakeType | VARCHAR | Loại phanh |
| Weight | DECIMAL | Trọng lượng xe (kg) |
| Color | VARCHAR | Màu sắc |
| WarrantyPeriod | INT(6) | Thời gian bảo hành (tháng) |

## **CLASS DIAGRAM**



1. **CHỨC NĂNG TỰ ĐỘNG**
2. **Server**
   1. **ConfigController**
      1. **Lấy tất cả thông tin của website**
         1. **Mô tả**

* Đây là chức năng lấy tất cả thông tin của website
  + - 1. **Cách hoạt động**
* Không có tham số
* Sử dụng phương thức findAll()
* Trả về là json chứa mảng config có các thông tin của website:id, website, saveFolder, scrapeTimes, keyword
  + - 1. **Quy trình**
* Gọi GET API: <http://192.168.101.7:8080/api/config/findAll>
  + 1. **Lấy website thông qua keyword**
       1. **Mô tả**
* Đây là chức năng lấy ra thông tin của website thông qua keyword
  + - 1. **Cách hoạt động**
* Tham số được truyền dưới dạng PathVariable, kiểu dữ liệu là String
* Sử dụng phương thức findByKeyWord(keyword)
* Trả về là json chứa mảng config có các thông tin của website:id, website, saveFolder, scrapeTimes, keyword
  + - 1. **Quy trình**
* Gọi GET API: [http://192.168.101.7:8080/api/config/get/{key}](http://192.168.101.7:8080/api/config/get/%7bkey%7d)
  + 1. **Thêm một website vào config**
       1. **Mô tả**
* Đây là chức năng thêm 1 website
  + - 1. **Cách hoạt động**
* Tham số được truyền dưới dạng RequestBody, là json config gồm các thông tin: website, saveFolder, keyword
* Kiểm tra keyword đã tồn tại hay chưa
* Nếu đã tồn tại thì không thêm và thông báo thêm thất bại
* Nếu chưa thì thêm và thông báo thêm thành công
  + - 1. **Quy trình**
* Gọi POST API: <http://192.168.101.7:8080/api/config/add>
  + 1. **Tăng số lần lấy dữ liệu lên 1**
       1. **Mô tả**
* Đây là chức năng dùng để tăng số lần lấy dữ liệu lên 1
* Được dùng để theo dõi số lần lấy dữ liệu của website, giúp tiết kiệm tài nguyên
  + - 1. **Cách hoạt động**
* Tham số được truyền vào dưới dạng PathVariable, kiểu dữ liệu là int
* Lấy ra id của website dùng method findById(id)
* Cập nhập scrapeTimes lên 1
* Dùng method save() để lưu
* Trả về Config mới nhất
  + - 1. **Quy trình**
* Gọi PUT API: [http://192.168.101.7:8080/api/config/increase/{id}](http://192.168.101.7:8080/api/config/increase/%7bid%7d)
  1. **DateDimController**
     1. **Lấy ID của ngày hôm nay trong bảng dateDim**
        1. **Mô tả**
* Đây là chức năng lấy ra id của ngày hiện tại để thêm vào bảng logs
  + - 1. **Cách hoạt động**
* Tham số được truyền dưới dạng RequestBody, là json của dateDim gồm các thông tin: fullDate
* Sử dụng method findByFullDate(today).getDateSk() để lấy ra id
* Trả về là kiểu dữ liệu int id
  + - 1. **Quy trình**
* Gọi GET API: <http://192.168.101.7:8080/api/dateDim/id>
  1. **EmailController**
     1. **Gửi email**
        1. **Mô tả**
* Đây là chức năng dùng để thông báo đến email khi bắt đầu các trạng thái WAITING, RUNNING, COMPLETE và FAILED
  + - 1. **Cách hoạt động**
* Tham số được truyền dưới dạng RequestParam. String receiver, String subject, String message
* Khởi tạo đối tượng SimpleMailMessage message
* message.setTo(receiver) để thiết lập người nhận
* message.setSubject(subject) để thiết lập chủ để của email
* message.setText(content) để thiết lập nội dung
* message.setFrom(FROM\_EMAIL) để thiết lập người gửi
* sử dụng method send(message) để gửi email
* Trả về json email gồm các thông tin: sender, receviver, subject, content
  + - 1. **Quy trình**
* Gọi GET API: <http://192.168.101.7:8080/api/email/send>
  1. **LogController**
     1. **Thêm dữ liệu vào logs**
        1. **Mô tả**
* Đây là chức năng thêm log
  + - 1. **Cách hoạt động**
* Tham số được truyền dưới dạng RequestBody, là json log bao gồm các thông tin: message, quantity, timeStart, timeEnd, dateSk, statusId, websiteId
* Kiểm tra nếu trạng thái không phải WAITING thì sẽ thêm vô
* Nếu là WAITING thì kiểm tra đã tồn tại websiteId và statusId hay chưa. Nếu tồn tại thì không thêm và thông báo thêm thất bại. Nếu chưa thì thêm và thông báo thành công
* Trả về là json log bao gồm các thông tin: id, message, quantity, timeStart, timeEnd, dateSk, statusId, websiteId
  + - 1. **Quy trình**
* Gọi POST API: <http://192.168.101.7:8080/api/log/add>
  1. **StatusController**
     1. **Lấy thông tin của trạng thái thông qua tên**
        1. **Mô tả**
* Đây là chức năng lấy thông tin của trạng thái thông qua tên để thêm vào log
  + - 1. **Cách hoạt động**
* Tham số được truyền dưới dạng PathVariable, kiểu dữ liệu là String
* Sử dụng method findByName(name) để lấy ra đối tượng status
* Trả về là json status có các thông tin: id, name
  + - 1. **Quy trình**
* Gọi GET API: [http://192.168.101.7:8080/api/status/log/getStatusByName/{name}](http://192.168.101.7:8080/api/status/log/getStatusByName/%7bname%7d)
  + 1. **Thêm dữ liệu vào status**
       1. **Mô tả**
* Đây là chức năng thêm status
  + - 1. **Cách hoạt động**
* Tham số được truyền dưới dạng RequestBody, là json status gồm các thông tin: name
* Sử dụng method save(status) để thêm vào CSDL
* Trả về json status có các thông tin gồm: name
  + - 1. **Quy trình**
* Gọi PUT API: <http://192.168.101.7:8080/api/status/log/add>
  1. **BikeController**
     1. **Thêm dữ liệu vào bike**
        1. **Mô tả**
* Đây là chức năng thêm bike
  + - 1. **Cách hoạt động**
* Tham số truyền vào dưới dạng RequestBody, là json bike gồm các thông tin: id, name, price, priceSale, brand, color, size status, description\_part1, description\_part2, description\_part3, timeStartScrape, timeEndScrape, timeStartInsert
* Sử dụng method save(bike) để thêm vào CSDL
* Trả về json bike gồm các thông tin: id, name, price, priceSale, brand, color, size status, description\_part1, description\_part2, description\_part3, timeStartScrape, timeEndScrape, timeStartInsert
  + - 1. **Quy trình**
* Gọi POST API: <http://192.168.101.7:8080/api/bike/add>
  + 1. **Xóa tất cả dữ liệu của bảng bikes** 
       1. **Mô tả**
* Đây là chức năng xóa tất cả dữ liệu trong bảng bikes để chuẩn bị cho dữ liệu mới
  + - 1. **Cách hoạt động**
* Không có tham số
* Sử dụng method deleteAll() để xóa tất cả dữ liệu
* Trả về thông báo xóa thành công
  + - 1. **Quy trình**
* Gọi DELETE API: <http://192.168.101.7:8080/api/bike/deleteAll>

1. **Client**
   1. **Config chuẩn bị để lấy dữ liệu**
      1. **Mô tả**

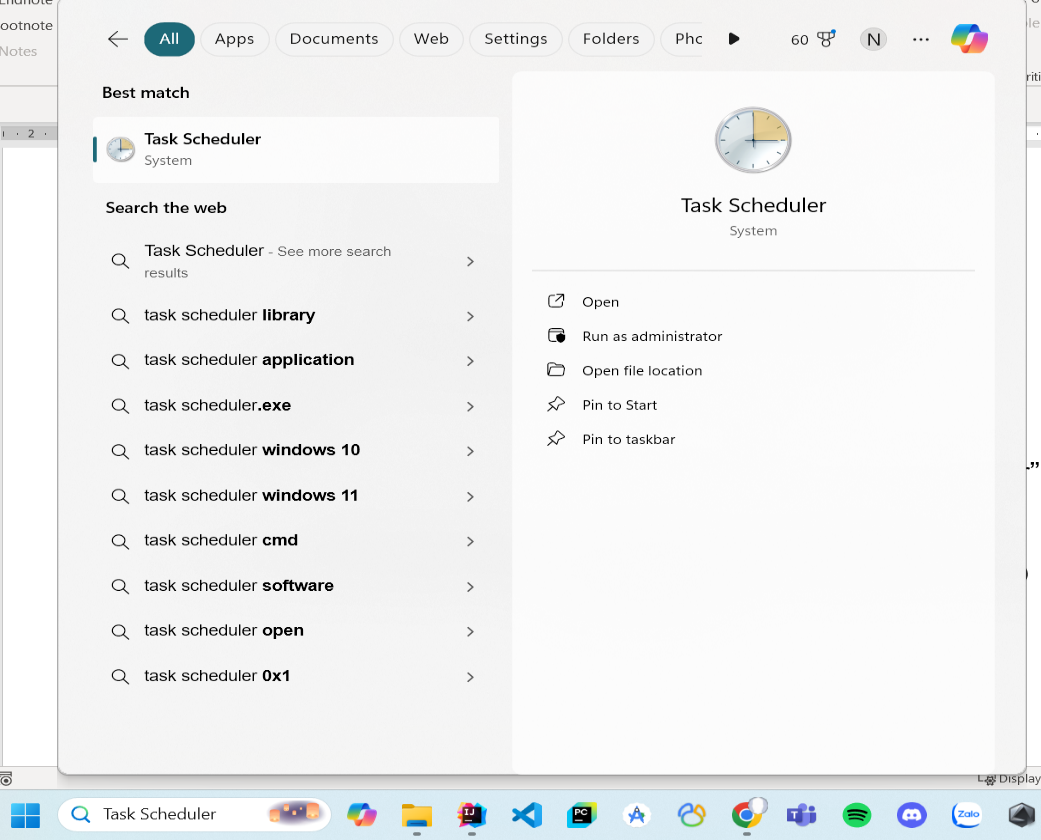
* Đây là chức năng để chuẩn bị cấu hình trước khi lấy dữ liệu từ các trang web
* Được tự động chạy vào lúc 00:00 mỗi ngày
* Cấu hình:
  + Thêm dữ liệu vào bảng logs cho các trang web ở trạng thái chờ(WAITING)
  + Xóa tất cả dữ liệu trong bảng bikes
  + Gửi email thông báo “Hệ thống đã sẵn sàng lấy dữ liệu của trang web”
    1. **Quy trình**
* Thêm dữ liệu vào logs
  + Gọi tới API: [http://192.168.101.7:8080/api/log/add](http://192.168.101.7:3306/api/log/add)
* Xóa tất cả dữ liệu trong bảng bikes
  + Gọi tới API: [http://192.168.101.7:8080/api/bike/deleteAll](http://192.168.101.7:3306/api/bike/deleteAll)
* Gửi email thông báo:
  + Gọi tới API: [http://192.168.101.7:8080/api/email/send](http://192.168.101.7:3306/api/email/send)
  1. **Lấy dữ liệu từ trang web bike2school.vn**
     1. **Mô tả**
* Đây là chức năng lấy dữ liệu từ trang web bike2school.vn
* Được chạy tự động lúc 01:00
  + 1. **Quy trình**
* Lấy các loại xe đạp của trang web
  + Lấy ra url của website trong CSDL
  + Vào website
  + Lấy ra các url lưu vào mảng pages
* Lấy url của tất cả số trang trong từng loại
  + Lặp qua mảng pages
  + Vô từng loại
  + Lấy ra url của từng số trang và lưu vào mảng urls
* Lấy tất cả sản phẩm trong tất cả trang
  + Lặp qua mảng urls
  + Vào từng số trang
  + Lấy url của tất cả sản phẩm trong từng trang
  + Lưu vào mảng hrefs
* Lấy dữ liệu
  + Lặp qua mảng hrefs
  + Vào từng trang của sản phẩm
  + Lấy ra các thông tin
* Lấy dữ liệu
  + Lặp qua mảng hrefs
  + Vào từng trang của sản phẩm
  + Lấy ra các thông tin
* Tổng hợp lại một quy trình lấy dữ liệu của trang web
  + lấy thông tin website
  + Gửi email thông báo “Hệ thống đang lấy dữ liệu của trang web”
  + Lấy ra id của datedim của ngày hôm nay
  + Thêm log với trạng thái bắt đầu lấy dữ liệu(RUNNING)
  + Lấy các loại xe đạp của trang web
  + Lấy url của tất cả số trang trong từng loại
  + Lấy tất cả sản phẩm trong tất cả trang
  + Lấy dữ liệu
  + Kiểm tra dữ liệu đã lấy
  + Nếu có dữ liệu thì thêm log thành công(COMPLETE) và gửi thông báo đến email “Hệ thống đã hoàn thành lấy dữ liệu của trang web”
  + Nếu không có dữ liệu thì thêm log thất bại(FAILED) và gửi thông báo đến email “Hệ thống lấy dữ liệu bị lỗi của trang web”
  1. **Lấy dữ liệu từ trang web xedapgiakho.com**
     1. **Mô tả**
* Đây là chức năng lấy dữ liệu từ trang web xedapgiakho.com
* Được chạy tự động lúc 01:00
  + 1. **Quy trình**
* Lấy các loại xe đạp của trang web
  + Lấy ra url của website trong CSDL
  + Vào website
  + Lấy ra các url lưu vào mảng pages
* Lấy url của tất cả số trang trong từng loại
  + Lặp qua mảng pages
  + Vô từng loại
  + Lấy ra url của từng số trang và lưu vào mảng urls
* Lấy tất cả sản phẩm trong tất cả trang
  + Lặp qua mảng urls
  + Vào từng số trang
  + Lấy url của tất cả sản phẩm trong từng trang
  + Lưu vào mảng hrefs
* Lấy dữ liệu
  + Lặp qua mảng hrefs
  + Vào từng trang của sản phẩm
  + Lấy ra các thông tin
* Tổng hợp lại một quy trình lấy dữ liệu của trang web
  + lấy thông tin website
  + Gửi email thông báo “Hệ thống đang lấy dữ liệu của trang web”
  + Lấy ra id của datedim của ngày hôm nay
  + Thêm log với trạng thái bắt đầu lấy dữ liệu(RUNNING)
  + Lấy các loại xe đạp của trang web
  + Lấy url của tất cả số trang trong từng loại
  + Lấy tất cả sản phẩm trong tất cả trang
  + Lấy dữ liệu
  + Kiểm tra dữ liệu đã lấy
  + Nếu có dữ liệu thì thêm log thành công(COMPLETE) và gửi thông báo đến email “Hệ thống đã hoàn thành lấy dữ liệu của trang web”
  + Nếu không có dữ liệu thì thêm log thất bại(FAILED) và gửi thông báo đến email “Hệ thống lấy dữ liệu bị lỗi của trang web”

1. **CÀI ĐẶT ĐỂ LẤY DỮ LIỆU TỰ ĐỘNG**
2. **Ứng dụng để chạy tự động**

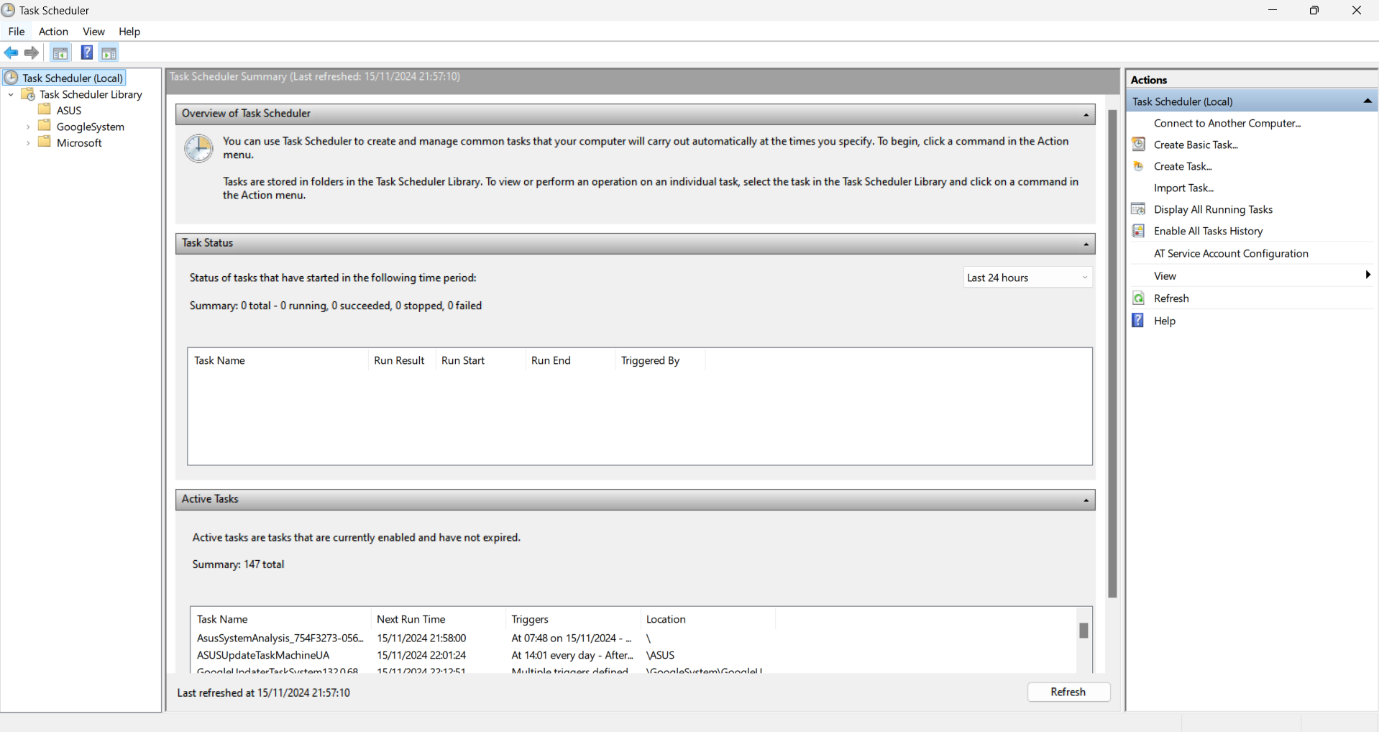
* Sử dụng “Task Scheduler” của window 11

1. **Cài đặt**

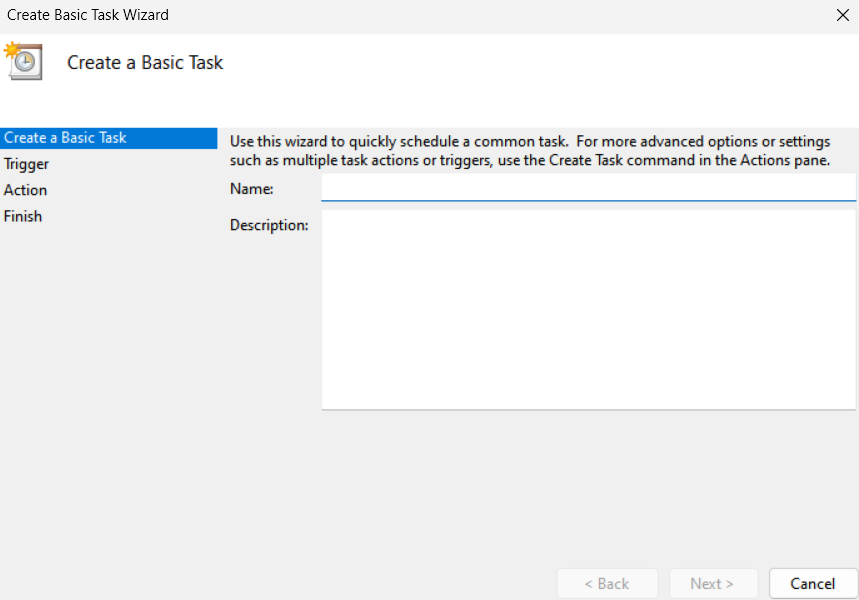
**B1:** Vào ô tìm kiếm của window nhập “Task Scheduler”



**B2:** Chọn “Create Basic Task”

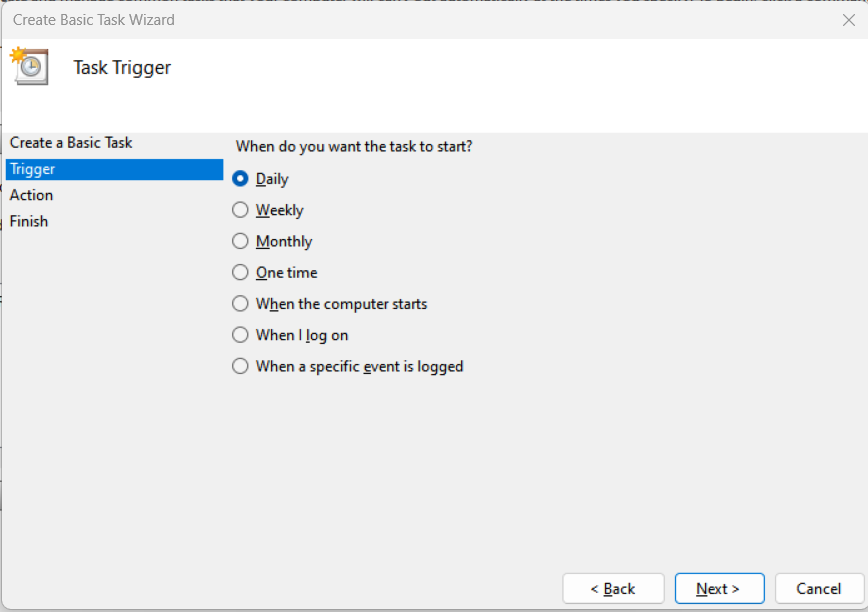


**B3:** Ở phần “Create a Basic Task”. Nhập “Name” và “Description” cho ứng dụng. Ấn “Next”

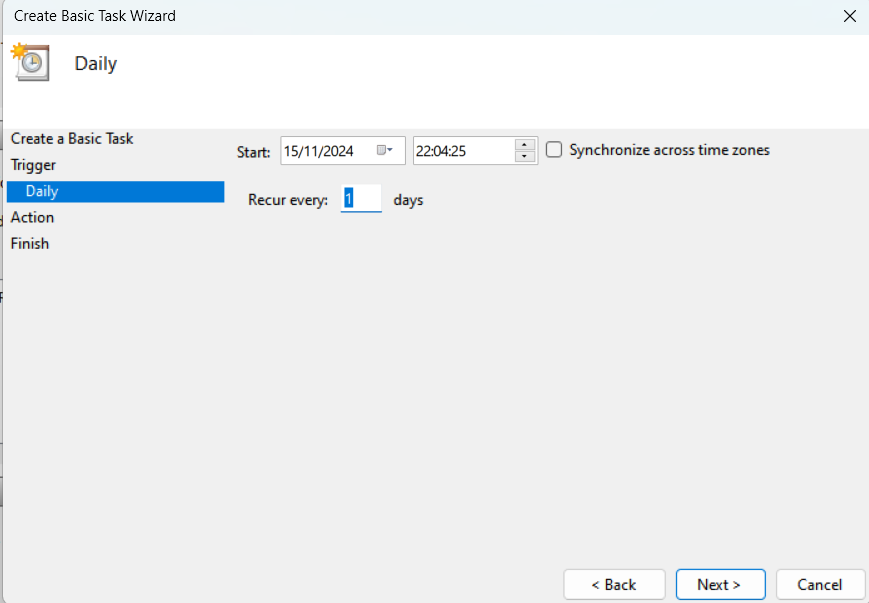


**B4:** Ở phần “Trigger”. Chọn chu kỳ lặp lại

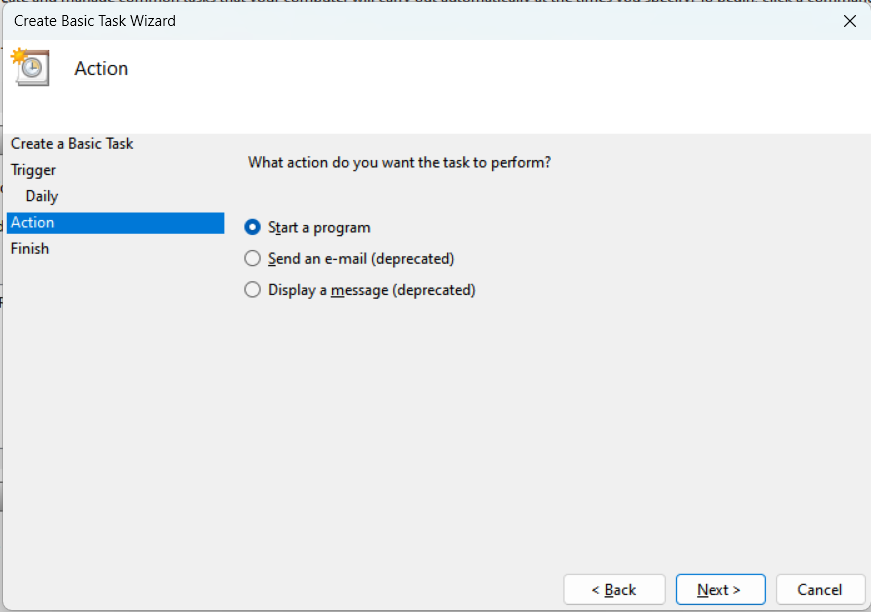
* + - “Daily”: hằng ngày
    - “Weekly”: hằng tuần
    - “Monthly”: Chạy vào các ngày cụ thể trong tháng, hoặc theo mẫu (ví dụ: ngày thứ N của tháng)
    - “One time”: Chạy một lần vào một thời điểm nhất định
* Ở đây chọn “Daily”. Ấn “Next”



**B5:** Ở phần “Trigger -> Daily”. Chọn ngày bắt đầu chạy, giờ bắt đầu và số lần lặp lại(Ví dụ chọn “Daily” thì mỗi 1 ngày 1 lần, …). Ấn “Next”

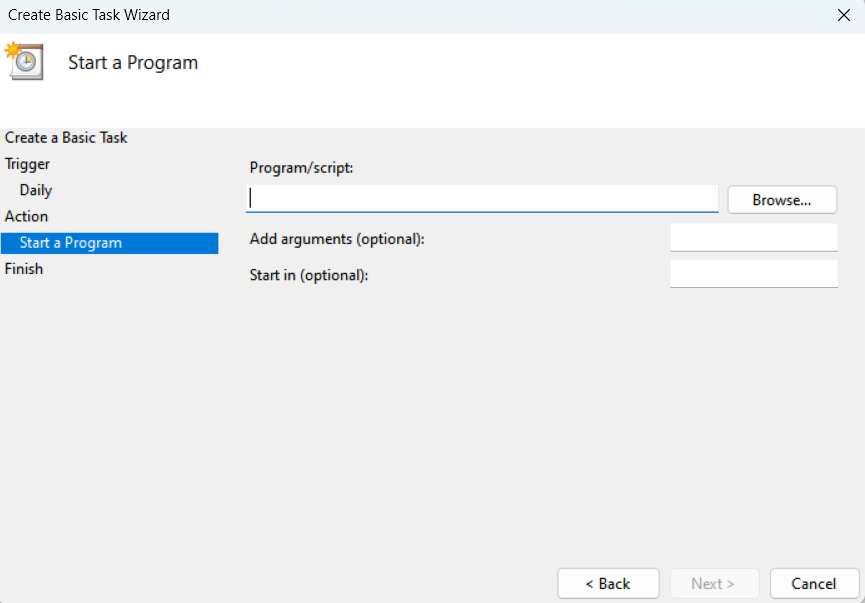
****

**B6:** Ở phần “Action”. Chọn “Start a program”. Ấn “Next”

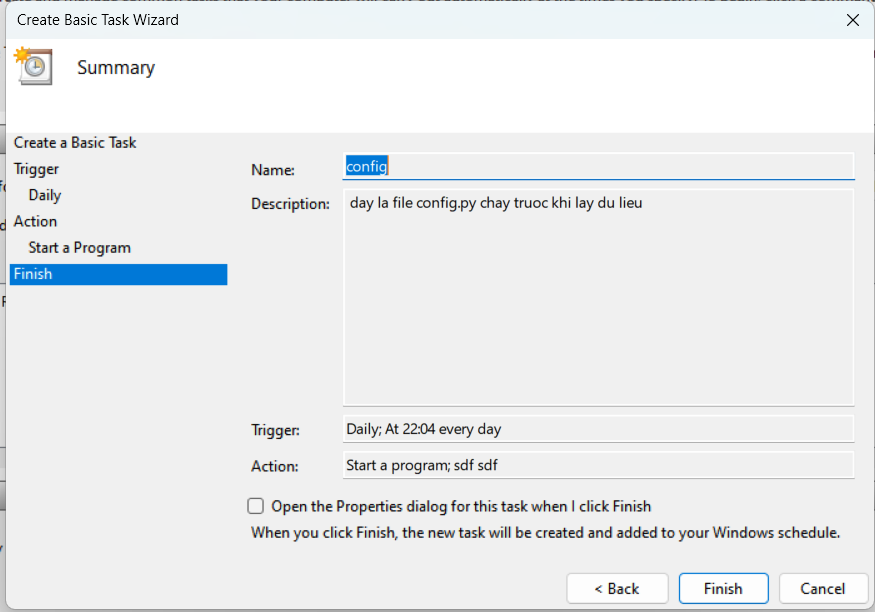
****

**B7:** Ở phần “Action -> Start a Program”

* Program/script: nhập đường dẫn cài đặt của python("D:\Python\Python312\python.exe")
* Add arguments(optional): nhập đường dẫn tới file python cần chạy
  + "D:\get\_data\_from\_web\config.py"
  + "D:\get\_data\_from\_web\scrape\_bike2school.py"
  + "D:\get\_data\_from\_web\scrape\_xedapgiakho.py"
* Start in (optional): nhập đường dẫn tới thư mục project python(D:\get\_data\_from\_web)
* Ấn “Next”

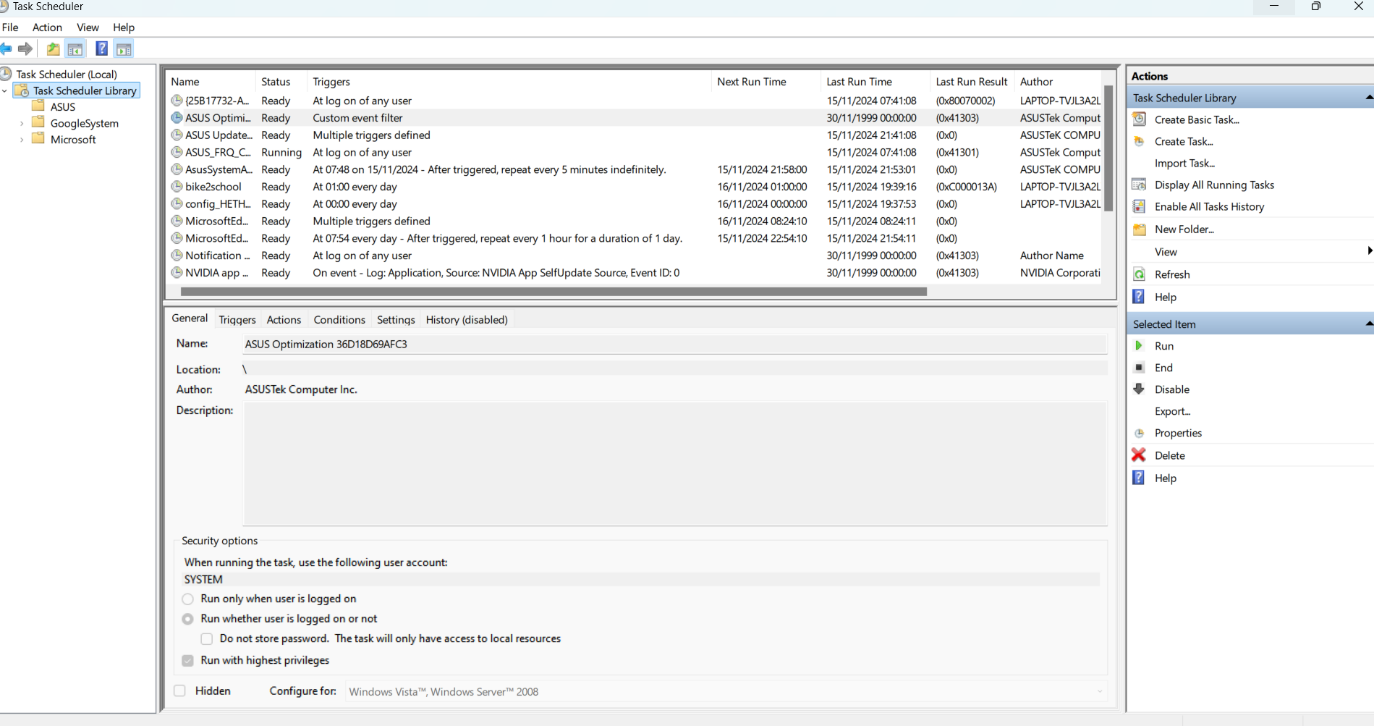


**B8:** Ở phần “Finish”. Xem lại thông tin đã thiết lập. Ấn “Finish” để hoàn tất

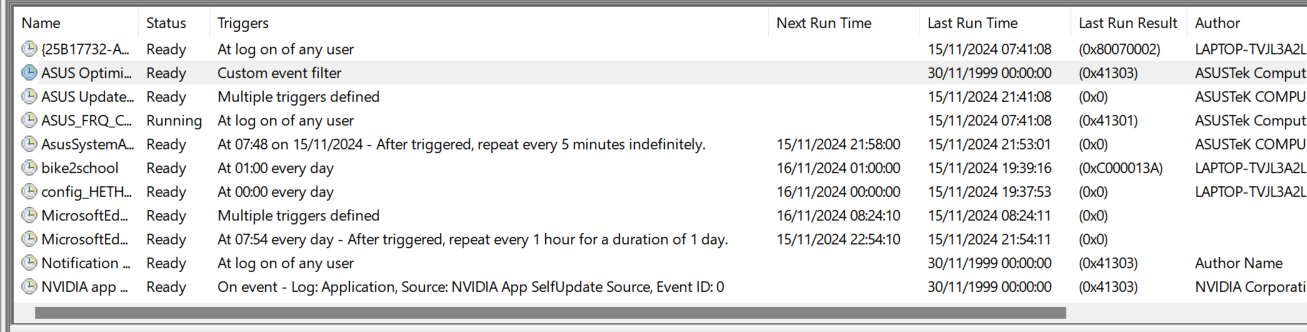
****

1. **Cấu hình chạy lại mỗi khi lỗi**

**B1:** Sau khi cài đặt xong, ở màn hình chính chọn “Task Scheduler Library”, tìm mục vừa cài đặt.

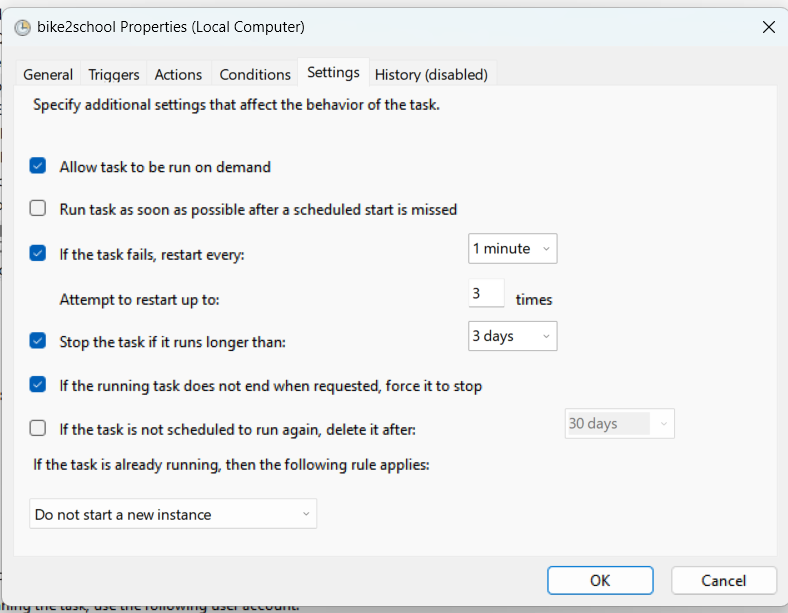
****

**B2:** Chọn “bike2school” vừa cài đặt

****

**B3:** Ở màn hình “bike2school Properties (Local Computer)”. Chọn “Settings”. Chọn ô “If the task fails, restart every”

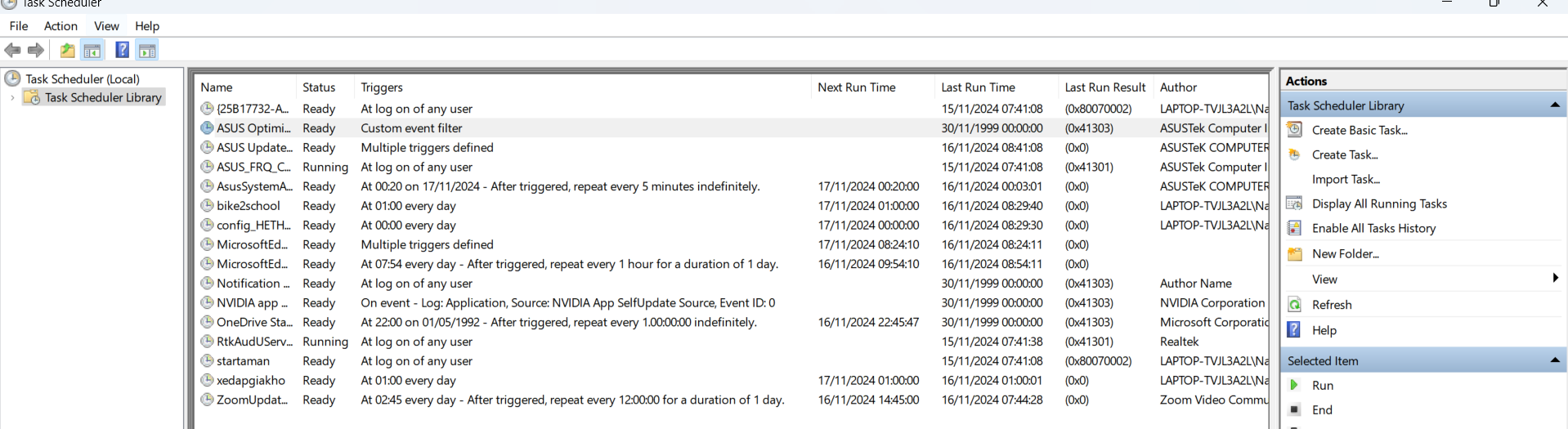
* Mặc định là lặp lại 3 lần cách nhau 1 phút



1. **HƯỚNG DẪN CHỨC NĂNG CHẠY BẰNG TAY**

**B1:** Vào ô tìm kiếm của window nhập “Task Scheduler”

**B2:** Chọn tên ứng dụng vừa cài đặt. Ở đây chọn “bike2school” hoặc “xedapgiakho” hoặc “config\_HETHONGBAOGIAXEDAP\_datawarehouse” và chọn “Run”



1. **SƠ ĐỒ LUỒNG THỰC HIỆN VIỆC LẤY DỮ LIỆU VÀO HỆ THỐNG**