**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**Logo, company name

Description automatically generated**

**KHÓA LUẬN TỐT NGHIỆP**

**XÂY DỰNG HỆ THỐNG WEBSITE QUẢN LÝ CỬA HÀNG ĐIỆN THOẠI BẰNG APACHE SPARK**

**Giảng viên hướng dẫn : THS. PHẠM VĂN ĐĂNG**

**Sinh viên thực hiện : ĐẶNG QUỐC LAI**

**Mã số sinh viên : 1811545103**

**Khóa : 2018**

**Chuyên ngành : Kỹ thuật phần mềm**

**TP. Hồ Chí Minh, tháng 1 năm 2022**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**Logo, company name

Description automatically generated**

**KHÓA LUẬN TỐT NGHIỆP**

**XÂY DỰNG HỆ THỐNG WEBSITE QUẢN LÝ CỬA HÀNG ĐIỆN THOẠI BẰNG APACHE SPARK**

**Giảng viên hướng dẫn : THS. PHẠM VĂN ĐĂNG**

**Sinh viên thực hiện : ĐẶNG QUỐC LAI**

**Mã số sinh viên : 1811545103**

**Khóa : 2018**

**Chuyên ngành : Kỹ thuật phần mềm**

**TP. Hồ Chí Minh, tháng 1 năm 2022**

**LỜI MỞ ĐẦU**

Ngày nay, khi công nghệ thông tin gần như xuất hiện ở mọi phương diện đời sống của con người, từ sinh hoạt cá nhân cho đến công việc cộng đồng thì một thiết bị điện thoại di động là vô cùng cần thiết. Điện thoại giúp con người liên lạc, trao đổi thông tin với nhau, giúp con người giải trí trong những lúc có thời gian rảnh,... Do đó, việc các cửa hàng bán điện thoại di động phát triển mạnh như hiện nay cũng không lấy gì làm lạ.

Tuy nhiên, kiểu bán hàng truyền thống gặp nhiều bất cập trong thời đại công nghệ bùng nổ như hiện nay, điển hình là việc các cửa hàng bán mĩ phẩm dần xuất hiện tràn lan trên các trang mạng xã hội. Và cửa hàng bán điện thoại cũng không ngoại lệ. Do đó, nhằm giải quyết các bất cập trên, các cửa hàng bán điện thoại đã nhanh chóng thích ứng bằng việc xây dựng một website bán điện thoại.

Bên cạnh việc phát triển thị trường thì việc lựa chọn sản phẩm kinh doanh cũng là một vấn đề khó đối với các nhà bán lẻ, việc lựa chọn dòng sản phẩm nào, số lượng bao nhiêu, vào giai đoạn nào là một vấn đề nan giải. Và giải pháp cho các vấn đề đó là thu thập dữ liệu từ các nhà bán lẻ khác để tiến hành đánh giá và nhập sản phẩm. Vì vậy, trong đồ án này, em mở rộng thêm việc sử dụng các công nghệ của Splash để thực hiện việc thu thập đó. Sau khi có được dữ liệu cần thiết, việc trực quan hóa dữ liệu cũng là một vấn đề không kém phần quan trọng, trực quan hóa dữ liệu giúp nhà kinh doanh dự đoán xu thế tiếp nhận sản phẩm của khách hàng đối với mặt hàng kinh doanh của mình.

LỜI CẢM ƠN

Em xin chân thành cảm ơn Giảng viên hướng dẫn ThS. Phạm Văn Đăng, người đã hướng dẫn tận tình, đóng góp các ý kiến chuyên môn cho báo cáo lần này. Thầy đã động viên tinh thần em cố gắng hoàn thành những nghiên cứu đặt ra. Thầy cũng cung cấp một số tài liệu liên quan đến đồ án mà tôi đang nghiên cứu và ân cần nhắc nhở tôi đến tiến độ thực hiện báo cáo này.

Tôi cũng chân thành gởi lời cảm ơn đến các Thầy, Cô khoa Công nghệ thông tin trường Đại học Nguyễn Tất Thành đã giảng dạy, hướng dẫn, trang bị các kiến thức cho tôi trong 3 năm học vừa qua, từ các kiến thức cơ bản đến các vấn đề chuyên sâu.

Tôi xin gởi lời cám ơn đến anh, chị, bạn bè, đồng nghiệp bằng nhiều hình thức khác nhau đã giúp đỡ tôi trong quá trình học tập tại trường cũng như trong thời gian hoàn thành báo cáo.

Tôi xin gởi lời cám ơn đến anh, chị, bạn bè, đồng nghiệp bằng nhiều hình thức khác nhau đã giúp đỡ tôi trong quá trình học tập tại trường cũng như trong thời gian hoàn thành báo cáo này.

Sinh viên thực hiện

*(Ký tên)*

**ĐẶNG QUỐC LAI**

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

TP. Hồ Chí Minh, ngày … tháng … năm 2022

**GIẢNG VIÊN HƯỚNG DẪN**

*(Ký tên)*

**ThS. Phạm Văn Đăng**

NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN

TP. Hồ Chí Minh, ngày … tháng … năm 2022

**GIẢNG VIÊN PHẢN BIỆN**

*(Ký, ghi rõ họ và tên)*

MỤC LỤC

[CHƯƠNG 1: GIỚI THIỆU ĐƠN VỊ THỰC TẬP 1](#_Toc89380491)

[1.1. Tổng quan về công ty 1](#_Toc89380492)

[1.2. Lịch sử hình thành 2](#_Toc89380493)

[1.3. Thông tin công ty 2](#_Toc89380494)

[1.4. Ngành nghề hoạt động 3](#_Toc89380495)

[1.5. Tầm nhìn và sứ mệnh 3](#_Toc89380496)

[CHƯƠNG 2: CỞ SỞ LÝ THUYẾT 4](#_Toc89380497)

[2.1. Version Source Control 4](#_Toc89380498)

[2.1.1. Tìm hiểu Version Source Control 4](#_Toc89380499)

[2.1.2. Tính hữu dụng của Version Source Control 5](#_Toc89380500)

[2.1.3. Git 5](#_Toc89380501)

[2.2. Database Management System MS-SQL 6](#_Toc89380502)

[2.2.1. Giới thiệu hệ quản trị cơ sở dữ liệu 6](#_Toc89380503)

[2.2.2. Kiểu dữ liệu trong MS-SQL 7](#_Toc89380504)

DANH MỤC BẢNG HÌNH

[Hình 1 Nguyên tắc truy xuất nguồn gốc 1](#_Toc89380447)

[Hình 2 Lịch sử hình thành công ty 2](#_Toc89380448)

[Hình 3 Minh họa Version Source Control 4](#_Toc89380449)

[Hình 4 Luồng công việc trong git 6](#_Toc89380450)

[Hình 5 Minh hoạt Data warehouse 12](#_Toc89380451)

[Hình 6 So sánh Data warehouse và Data lake 13](#_Toc89380452)

DANH MỤC BẢNG BIỂU

[Bảng 1 Ngành nghề hoạt động của công ty 3](#_Toc89380558)

[Bảng 2 Kiểu dữ liệu trong SQL 7](#_Toc89380559)

[Bảng 3 So sánh Clustered Index và Non-Clustered Index 9](#_Toc89380560)

[Bảng 4 So sánh bảng deleted và inserted 11](#_Toc89380561)

[Bảng 5 So sánh Data warehouse và Data lake 13](#_Toc89380562)

[Bảng 6 Ưu nhược điểm Cloud và On-Premises 14](#_Toc89380563)

[Bảng 7 Các loại Staging Layer 14](#_Toc89380564)

[Bảng 8 So sánh ETL và ELT 15](#_Toc89380565)

[Bảng 9 Các loại Slowly Changing Dimension 17](#_Toc89380566)

[Bảng 10 Các câu lệnh làm việc với Database và Collection 18](#_Toc89380567)

[Bảng 11 Một số toán tử trong MongoDB 19](#_Toc89380568)

[Bảng 12 Toán tử cập nhật dữ liệu 20](#_Toc89380569)

DANH SÁCH TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| STT | Từ viết tắt | Ý nghĩa |
| 1 | API | Application Programming Interface |
| 2 | AWS | Amazone Web Services |
| 3 | CSDL | Cơ sở dữ liệu |
| 4 | DAG | Directed Acyclic Graph |
| 5 | DBMS | Database Management System |
| 6 | ETL | Extract, Transform and Load |
| 7 | MS-SQL | Microsoft Structured Query Language |
| 8 | RDD | Resilient Distributed Datasets |
| 9 | UDF | User-Defined Functions |
| 10 | VSC | Version Source Control |

|  |  |
| --- | --- |
| Trường Đại học Nguyễn Tất Thành  **Khoa Công Nghệ Thông Tin**  🙜 🙜 🙝 🙝 | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  **Độc lập – Tự do – Hạnh phúc**  🙜 🙜 🙝 🙝 |

**NHIỆM VỤ KHÓA LUẬN TỐT NGHIỆP**

*CHÚ Ý: Sinh viên phải đóng tờ này vào trang thứ nhất của bản thuyết minh*

Họ và tên : **ĐẶNG QUỐC LAI** MSSV : **1811545103** Chuyên ngành : **Kỹ thuật phần mềm** Lớp : **18DTH2A**

1. Tên đề tài:

**Xây dựng hệ thống website quản lý cửa hàng điện thoại bằng Apache Spark.**

1. Giáo viên hướng dẫn: **ThS. Phạm Văn Đăng**
2. Nhiệm vụ / nội dung đề tài (mô tả chi tiết nội dung, yêu cầu, phương pháp… ):
   * + Khảo sát hiện trạng.
     + Phân tích và thiết kế.
     + Xây dựng cơ sở dữ liệu.
     + Xây dựng hệ thống thu thập dữ liệu.
     + Thiết kế và cài đặt Website:
       - Trang chủ.
       - Trang chi tiết.
       - Trang bình luận.
     + Tải dự liệu vào cơ sở dữ liệu bằng Apache Spark.
     + Trực quan hóa dữ liệu thu thập được.
     + Dự đoán xu hướng mua hàng từ dữ liệu.
     + Viết báo cáo theo mẫu qui định.
     + Làm file powerpoint thuyết trình
3. Thời gian thực hiện từ: 10/10/2021 đến: 15/01/2021
4. Nội dung và yêu cầu Khóa luận tốt nghiệp đã được thông qua Bộ môn.

|  |  |
| --- | --- |
| **TRƯỞNG BỘ MÔN**  *(Ký và ghi rõ họ tên)*  **ThS.Phạm Văn Đăng** | *Ngày 03 tháng 10 năm 2021*  **GIÁO VIÊN HƯỚNG DẪN**  *(Ký và ghi rõ họ tên)*  **ThS. Phạm Văn Đăng** |

# GIỚI THIỆU

* 1. **Giới thiệu**

Ngày nay, một chiếc điện thoại như là vật bất ly thân của mọi người, điện thoại không chỉ giúp chúng ta giao tiếp, truyền tải thông tin với những người xung quanh mà còn giúp chúng ta giải trí, nghe nhạc, lướt facebook, nhắn tin với bạn bè,… Vì vậy mà các cửa hàng, trung tâm bán điện thoại ngày này ngày càng phát triển. Tuy nhiên, ở các cửa hàng truyền thống có một số vấn đề như sau: cửa hàng chỉ mở cửa ban ngày, một số có mở cửa thêm vào giờ đêm nhưng thường tới 20 – 21 giờ, điều đó gây cản trở cho một số nhóm khách hàng vì ban ngày phải đi làm việc và khó có thời gian tới cửa hàng để xem và chọn mua sản phẩm. Cửa hàng nhập sản phẩm mới tuy nhiên lại không thể thông báo tới khách hàng, khó tiếp cận với khách hàng; Có nhiều khách hàng muốn tìm hiểu sản phẩm nhưng lại ngại khi ra cửa hàng,…

Để giải quyết được các vấn đề trên thì việc cho ra đời một website bán điện thoại di động là hoàn toàn hợp lý. Ngoài các bất cập nêu trên, website còn giúp tương tác giữa nhân viên cửa hàng với khách hàng, phục vụ khách hàng nhiều hơn. Về phía khách hàng của cửa hàng, khách hàng có thể vào xem sản phẩm, yêu cầu nhân viên tư vấn, hỗ trợ,…

Website không chỉ giúp chạy trên nhiều nền tảng như máy tính để bàn, máy tính xách tay, điện thoại, thậm chí các sản phẩm tích hợp công nghệ hiện đại như đồng hồ thông minh, TV thông minh,… mà còn giúp việc quảng bá sản phẩm tốt hơn thông qua việc kết nối với các nền tảng mạng xã hội ngày nay, tiếp cận với nhiều nhóm khách hàng hơn. Không chỉ vậy, chi phí để xây dựng và phát triển một website rẻ hơn so với việc tạo ra các banner, các quảng cáo trên truyền hình.

* 1. **Khảo sát hiện trạng**

Khi đến các chi nhánh cửa hàng Thế giới di động, người dùng sẽ thấy các sản phẩm được đặt trên kệ, các sản phẩm được sắp xếp theo thứ tự về giá, về mức độ tiêu thụ hoặc về yêu cầu của khách hàng (ví dụ khách hàng là người dùng phổ thông sẽ đến phần các điện thoại phổ thông; người dùng sử dụng điện thoại phục vụ nhu cầu chơi game giải trí sẽ đến kệ đặt các điện thoại có cấu hình cao; người dùng sử dụng điện thoại với mục đích chụp ảnh sẽ đến các quầy điện thoại có camera tốt,...). Các sản phẩm được trình bày bắt mắt, sản phẩm ở trên, giá ở dưới, kèm theo các khuyến mãi đi kèm. Khách hàng muốn mua sản phẩm chỉ việc tới yêu cầu nhân viên gần mình nhất và đợi thanh toán là có thể mang sản phẩm ra về.

Đồng thời, khách hàng có thể chọn nhiều sản phẩm cùng lúc, yêu cầu thanh toán tại chỗ hoặc thanh toán rồi yêu cầu giao hàng vào một thời điểm khác. Hơn nữa, khi đến với trung tâm, khách không chỉ mua điện thoại mà ngay cả các thiết bị hỗ trợ như tai nghe, thẻ nhớ, chuột máy tính,... cũng là các mặt hàng được ưa chuộng. Khách hàng thanh toán có thể sử dụng nhiều hình thức: tiền mặt, thẻ, mã giảm giá,...

* 1. **Mô tả bài toán**

Đến với trung tâm Thế giới di động, người dùng có thể trải nghiệm trực tiếp được sản phẩm, đặt mua sản phẩm và thanh toán ngay lập tức mà không cần phải chờ đợi giao hàng, chờ đợi xác thực thông tin,... Đặc biệt, trung tâm còn có chính sách khuyến mãi đi kèm khi mua hàng tại đây, ví dụ: khách hàng có thể sử dụng mã giảm giá sau khi thanh toán để mua hàng tiếp ở Bách hóa xanh, Điện máy xanh,... Ngoài ra, chính sách bảo hành cũng là điểm cộng trong mắt khách hàng về Thế giới di động bởi việc chi nhánh rộng khắp cả nước, khách hàng có thể mang sản phẩm và bảo hành tại bất kỳ chi nhánh nào và đặt điểm nhận tại chi nhánh khác (nếu muốn).

Tuy nhiên, việc khách hàng phải đến cửa hàng trực tiếp có nhiều bất cập, ví dụ như đối với khách hàng là doanh nhân, họ có hiếm hoặc rất ít thời gian để đến đây, trải nghiệm sản phẩm và mua sắm. Trong khi doanh nhân là tầng lớp sử dụng thiết bị công nghệ rất nhiều, đặc biệt, công nghệ với họ phải đảm bảo, chất lượng, giá thành không thành vấn đề. Hay như trong tình trạng dịch bệnh Covid-19 năm nay, việc ra đường hoặc qua lại những nơi đông người là điều kiêng kỵ thì việc mua sắm online lại trở nên cần thiết.

Vì vậy, để giải quyết các vấn đề trên, một cửa hàng, trung tâm bán điện thoại cần có nhất là một website để có thể kết nối với nhiều tầng lớp người dùng. Một website còn giúp chủ cửa hàng quản lý các sản phẩm, hiển thị đề xuất nhập sản phẩm hoặc kiểm tra các đơn hàng trong kho của mình.

* 1. **Các đối tượng quản lý**
* Khách hàng.
* Nhân viên.
* Giỏ hàng.
* Đơn đặt hàng.
* Điện thoại.
* Doanh thu, lợi nhuận.
* Phụ kiện điện thoại.
* Khuyến mãi đi kèm.
  1. **Mô tả các đối tượng quản lý**
* Khách hàng: là tập hợp những cá nhân, nhóm người, tổ chức, doanh nghiệp… có nhu cầu sử dụng sản phẩm của cửa hàng.
* Nhân viên: là người lao động, chịu sự quản lý, giám sát của chủ cửa hàng. Đồng thời thực hiện các công việc bao gồm: quản lý sản phẩm, thêm, xóa, sửa thông tin sản phẩm, giao tiếp, hỗ trợ với khách hàng khi có yêu cầu.
* Giỏ hàng: là đối tượng được lưu ở bộ nhớ cục bộ của máy tính khách hàng, sẽ được cập nhật vào CSDL sau khi khách hàng đã đăng nhập và đóng trang web. Tuy nhiên, trường hợp khách hàng đóng trang web trong khi chưa đăng nhập thì giỏ hàng vẫn được lưu ở bộ nhớ cục bộ, nhưng nếu chuyển sang máy tính khác thì thông tin giỏ hàng sẽ không còn.
* Đơn đặt hàng: là trạng thái giỏ hàng được gửi cho nhân viên và đã được nhân viên tiếp nhận (trường hợp nhân viên chưa tiếp nhận thì sẽ ở giỏ hàng). Đơn đặt hàng chứa các thông tin cần thiết như: tên khách hàng, ngày đặt, ngày giao, số điện thoại khách hàng, điện thoại, địa chỉ nhận hàng, thông tin đơn hàng, giá tiền, tổng tiền, ghi chú (nếu có)...
* Điện thoại: là sản phẩm điện thoại của cửa hàng đang bày bán. Điện thoại bao gồm các thông tin như: Mã điện thoại, tên điện thoại, tên nhà sản xuất, tên phiên bản, hình ảnh minh họa, năm phát hành, cấu hình chi tiết (ram, rom, camera, chip, kích thước, trọng lượng,...). Ngoài ra, còn phải quản lý số lượng tồn kho của sản phẩm đó ở các chi nhánh để kịp thời điều chỉnh, bổ sung, nhập hàng hoặc mượn hàng từ các chi nhánh khác (ví dụ khi khách hàng yêu cầu sản phẩm ở 1 chi nhánh để ra đó lấy, nhưng tại chi nhánh đó, sản phẩm đó hiện tại không còn, thì chi nhánh này sẽ tiến hành mượn sản phẩm từ chi nhánh khác qua).
* Doanh thu, lợi nhuận: là lãi sinh ra khi sản phẩm được bán ra và trừ đi các chi phí như mặt bằng, lương nhân viên, phí sinh hoạt khác của chi nhánh,... Doanh thu quyết định sự tồn tại của chi nhánh, ví dụ 1 chi nhánh của Thế giới di động vì không đạt doanh thu trong 1 thời gian dài thì sẽ bị đóng cửa.
* Phụ kiện điện thoại là các linh kiện đi kèm theo điện thoại ví dụ như tai nghe, loa, miếng dán màn hình,... Ngoài điện thoại là mặt hàng chính thì phụ kiện điện thoại cũng được xem như một sản phẩm mang lại lợi nhuận cho cửa hàng.
* Khuyến mãi đi kèm là các chương trình khuyến mãi đi kèm theo sản phẩm tại các chi nhánh khác nhau hoặc toàn bộ chi nhánh. Khuyến mãi nhằm giúp tiêu thụ sản phẩm nhanh hơn đặc biệt là vào các thời điểm đặc biệt trong năm như Black Friday, Quốc khánh (2/9),...
  1. **Các yêu cầu chức năng**
* Đối với khách hàng:
  + Đăng nhập.
  + Đăng ký.
  + Thay đổi thông tin tài khoản.
  + Xem danh sách sản phẩm.
  + Tìm kiếm sản phẩm.
  + Thêm sản phẩm vào giỏ hàng.
  + Yêu cầu đặt hàng, chọn phương thức thanh toán.
  + Tra cứu đơn hàng.
  + Tra cứu thời hạn bảo hành của sản phẩm.
* Đối với nhân viên:
  + Đăng nhập.
  + Xem, thêm, xóa, sửa thông tin sản phẩm.
  + Tiếp nhận, phản hồi đơn hàng từ khách hàng.
  + Phản hồi lại các bình luận của khách hàng về sản phẩm.
  + Trợ giúp đối với khách hàng có yêu cầu cần hỗ trợ.
  1. **Phạm vi của đề tài**

Giới hạn của hệ thống nằm trong các chức năng:

* Xem sản phẩm.
* Đăng ký, đăng nhập.
* Xem giỏ hàng.
* Xem trạng thái đơn hàng.

1. **CƠ SỞ LÝ THUYẾT**
   1. **React JS**
      1. **Giới thiệu**

React JS là một thư viện UI phát triển tại Facebook để hỗ trợ việc xây dựng những thành phần (components) UI có tính tương tác cao, có trạng thái và có thể sử dụng lại được. React được sử dụng tại www.facebook.com và www.instagram.com được viết hoàn toàn trên React. React JS đang nổi lên trong những năm gần đây với xu hướng Single Page Application. Trong khi những framework khác cố gắng hướng đến một mô hình MVC hoàn thiện thì React nổi bật với sự đơn giản và dễ dàng phối hợp với những thư viện Javascript khác. React JS tích hợp giữa Javascript và HTML vào trong JSX làm cho các component dễ hiểu hơn.

* + 1. **Virtual DOM**

Công nghệ Virtual DOM giúp tăng hiệu năng cho ứng dụng. Việc chỉ node gốc mới có trạng thái và khi nó thay đổi sẽ tái cấu trúc lại toàn bộ, đồng nghĩa với việc DOM tree cũng sẽ phải thay đổi một phần, điều này sẽ ảnh hưởng đến tốc độ xử lý. React JS sử dụng Virtual DOM (DOM ảo) để cải thiện vấn đề này.Virtual DOM là một object Javascript, mỗi object chứa đầy đủ thông tin cần thiết để tạo ra một DOM, khi dữ liệu thay đổi nó sẽ tính toán sự thay đổi giữa object và tree thật, điều này sẽ giúp tối ưu hoá việc re-render DOM tree thật.

React dùng để xây dựng các ứng dụng lớn mà dữ liệu của chúng thay đổi liên tục theo thời gian. Dữ liệu thay đổi thì hầu hết kèm theo sự thay đổi về giao diện. Ví dụ như tin nhắn trên Facebook: khi một người dùng thực hiện việc gửi tin nhắn, thì tin nhắn đó cần cập nhật ở cả 2 người dùng, đồng thời, giao diện phải reload lại toàn bộ trang mới có thể tải tin nhắn mới, nhưng với react, web sẽ nhận biết có tin nhắn mới hay không? Nếu có thì chỉ cần render mỗi 1 tin nhắn đó thôi, các tin nhắn còn lại giữ nguyên.

* + 1. **JavaScript XML (JSX)**

JSX là một loại cú pháp mở rộng dành cho ngôn ngữ JavaScript viết theo kiểu XML. JSX cung cấp cú pháp ngọt (syntactic sugar) để thay cho câu lệnh React.create-Element() trong React. Mã lệnh viết bằng JSX sẽ được chuyển sang JavaScript để trình duyệt có thể hiểu được.

Trên thực tế các trình duyệt ngay cả các trình duyệt mới nhất cũng không hỗ trợ cú pháp của JSX. Do đó source code viết sử dụng JSX cần được biên soạn về JavaScript sử dụng một thư viện có tên là Babel.

Đặc điểm của JSX:

* Faster: Nhanh hơn. JSX thực hiện tối ưu hóa trong khi biên dịch sang mã Javacsript. Các mã này cho thời gian thực hiện nhanh hơn nhiều so với một mã tương đương viết trực tiếp bằng Javascript.
* Safer: An toàn hơn. Ngược với Javascript, JSX là kiểu statically-typed, nghĩa là nó được biên dịch trước khi chạy, giống như Java, C++. Vì thế các lỗi sẽ được phát hiện ngay trong quá trình biên dịch. Ngoài ra, nó cũng cung cấp tính năng gỡ lỗi khi biên dịch rất tốt.
* Easier: Dễ dàng hơn. JSX kế thừa dựa trên Javascript, vì vậy rất dễ dàng để cho các lập trình viên Javascripts có thể sử dụng.
  + 1. **Component**

React được xây dựng xung quanh các component, không dùng template như các framework khác. Trong React, chúng ta xây dựng trang web sử dụng những thành phần (component) nhỏ. Chúng ta có thể tái sử dụng một component ở nhiều nơi, với các trạng thái hoặc các thuộc tính khác nhau, trong một component lại có thể chứa thành phần khác. Mỗi component trong React có một trạng thái riêng, có thể thay đổi, và React sẽ thực hiện cập nhật component dựa trên những thay đổi của trạng thái. Mọi thứ React đều là component. Chúng giúp bảo trì mã code khi làm việc với các dự án lớn. Một react component đơn giản chỉ cần một method render. Có rất nhiều methods khả dụng khác, nhưng render là method chủ đạo, giống như main method trong Java và C.

* + 1. **Props**

Props: giúp các component tương tác với nhau giống như việc truyền tham số trong các ngôn ngữ lập trình khác, component nhận input gọi là props, và trả thuộc tính mô tả những gì component con sẽ render. Prop là bất biến.

* + 1. **State**

State: thể hiện trạng thái của ứng dụng, khi state thay đồi thì component đồng thời render lại để cập nhật giao diện.

* + 1. **React Router DOM**

React Router là một thư viện định tuyến (routing) tiêu chuẩn trong React. Nó giữ cho giao diện của ứng dụng đồng bộ với URL trên trình duyệt. React Router cho phép bạn định tuyến "luồng dữ liệu"(data flow) trong ứng dụng của bạn một cách rõ ràng. Ứng với một URL, nó sẽ tương ứng với một Route, mỗi Route lại tương ứng với một Component và render ra một giao diện khác nhau.

* + 1. **Bootstrap**

Bootstrap là một bộ sưu tập miễn phí của các mã nguồn mở và công cụ dùng để tạo ra một mẫu webiste hoàn chỉnh. Với các thuộc tính về giao diện được quy định sẵn như kích thước, màu sắc, độ cao, độ rộng…

Một số điểm nổi trội của Bootstrap:

* Dễ dàng thao tác: cơ chế hoạt động của Bootstrap là dựa trên xu hướng mã nguồn mở HTML, CSS và Javascript. Người dùng cần trang bị kiến thức cơ bản 3 mã này mới có thể sử dụng Bootstrap hiệu quả. Bên cạnh đó, các mã nguồn này cũng có thể dễ dàng thay đổi và chỉnh sửa tùy ý.
* Tùy chỉnh dễ dàng: Bootstrap được tạo ra từ các mã nguồn mở cho phép designer linh hoạt hơn. Giờ đây có thể lựa chọn những thuộc tính, phần tử phù hợp với dự án họ đang theo đuổi.
* Độ tương thích cao: Điểm cộng lớn nhất của Bootstrap là khả năng tương thích với mọi trình duyệt và nền tảng. Đây là một điều cực kì quan trọng và cần thiết trong trải nghiệm người dùng. Sử dụng Grid System cùng với hai bộ tiền xử lý Less và Sass, Bootstrap mặc định hỗ trợ Responsive và ưu tiên cho các giao diện trên thiết bị di động hơn. Bootstrap có khả năng tự động điều chỉnh kích thước trang website theo khung browser. Mục đích để phù hợp với màn hình của máy tính để bàn, tablet hay laptop.
  + 1. **React Slick**

Slick.js là một plugin jQuery nổi tiếng được tạo ra bởi Ken Wheeler cho phép bạn tạo ra những responsive carousel tuyệt đẹp. Tuy nhiên, khi chúng ta sử dụng React, được khuyến nghị không nên sử dụng jQuery, vì thế thư viện React Slick ra đời nhằm hỗ trợ React mà không cần jQuery.

* 1. **Node JS**

NodeJS được phát triển bởi Ryan Dahl vào năm 2009 và có thể chạy trên nhiều hệ điều hành khác nhau: OS X, Microsoft Windows, Linux.

NodeJS là một nền tảng được xây dựng trên V8 JavaScript Engine – trình thông dịch thực thi mã JavaScript, giúp xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở rộng.

Đặc điểm của NodeJS

* NodeJS được viết bằng JavaScript với cộng đồng người dùng lớn mạnh.
* Tốc độ xử lý nhanh: nhờ cơ chế xử lý bất đồng độ (non-blocking), NodeJS có thể xử lý hàng ngàn kết nối cùng lúc mà không gặp bất cứ khó khăn nào.
* Dễ dàng mở rộng: dùng để phát triển website.
  1. **PostgreSQL**
     1. **Giới thiệu PostgreSQL**

PostgreSQL là một hệ thống quản trị cơ sở dữ liệu quan hệ - đối tượng (object-relational database management system) có mục đích chung, hệ thống cơ sở dữ liệu mã nguồn mở tiên tiến nhất hiện nay.

PostgreSQL được thiết kế để chạy trên các nền tảng tương tự UNIX. Tuy nhiên, PostgreSQL sau đó cũng được điều chỉnh linh động để có thể chạy được trên nhiều nền tảng khác nhau như Mac OS X, Solaris và Windows.

PostgreSQL là một phần mềm mã nguồn mở miễn phí. Mã nguồn của phần mềm khả dụng theo license của PostgreSQL, một license nguồn mở tự do. Theo đó, bạn sẽ được tự do sử dụng, sửa đổi và phân phối PostgreSQL dưới mọi hình thức.

PostgreSQL không yêu cầu quá nhiều công tác bảo trì bởi có tính ổn định cao. Do đó, nếu bạn phát triển các ứng dụng dựa trên PostgreSQL, chi phí sở hữu sẽ thấp hơn so với các hệ thống quản trị dữ liệu khác.

* + 1. **Đặc điểm PostgreSQL**

PostgreSQL là hệ thống quản lý cơ sở dữ liệu đầu tiên triển khai tính năng kiểm soát đồng thời nhiều phiên bản (MVCC) trước cả Oracle. Tính năng MVCC cũng tương tự với các snapshot riêng biệt trong Oracle.

Là hệ thống quản lý cơ sở dữ liệu quan hệ đối tượng, PostgreSQL cho phép thêm vào các tính năng tùy chỉnh được phát triển bằng các ngôn ngữ chương trình khác nhau như C/C , Java,...

Trên thị trường, có nhiều công ty đã xây dựng và cung cấp các sản phẩm, giải pháp sử dụng PostgreSQL. Một số công ty nổi bật trong số đó là Apple, Fujitsu, Red Hat, Cisco, Juniper Network,...

## Web Scraping

### Regex

Regex là các mẫu (pattern) thay vì các chuỗi cụ thể được sử dụng tìm/thay thế (Find/Replace). Là một công cụ cực mạnh cho xử lí chuỗi trong Python… Ví dụ: Khi kiểm tra tính hợp lệ của email hoặc số điện thoại thì điều đầu tiên chính là regex. Regex là viết tắt của Regular Expression, tên thuần Việt là biểu thức chính quy.

Một số biểu thức trong regex:

* **[xyz]** Tìm và so sánh tất cả ký tự nằm trong dấu ngoặc vuông và trùng khớp với 1 ký tự trong dấu ngoặc vuông.
* **[a-z]** So sánh và trùng khớp với một ký tự nằm trong khoảng chỉ định.
* **[^xyz]** So sánh và không trùng khớp với những ký tự nằm trong khoảng chỉ định. Dấu ^ (dấu mũ) nằm trong dấu ngoặc vuông là một dấu phủ định.
* **^** Trùng khớp với phần đầu của chuỗi đích.
* **$** Trùng khớp với phần cuối của chuỗi đích.
* **+** Trùng khớp với 1 hoặc nhiều lần ký tự đứng trước nó.
* **?** Trùng khớp với 0 hoặc 1 lần ký tự đứng trước nó.
* **.** Trùng khớp với 1 ký tự đơn bất kỳ ngoại trừ ký tự ngắt dòng (line-break) và cũng không lấy được ký tự có dấu (unicode).
* **x{n}** Trùng khớp đúng với n lần ký tự đứng trước nó. n là một số không âm.
* **x{n,}** Trùng khớp với ít nhất n lần ký tự đứng trước nó. n là một số không âm.
* **x{n,m}** Trùng khớp với ít nhất n lần và nhiều nhất là m lần ký tự đứng trước nó. n và m là một số không âm và n <= m.
* **x|y** Trùng khớp với x hoặc y.

### Xây dựng một Spider

* Khởi tạo một project: scrapy startproject stack
* Tạo một spider: scrapy genspider name example\_domain.com

### Debug trong Spider

Debug là quá trình tìm kiếm ra lỗi hay nguyên nhân gây ra lỗi để có hướng sửa lỗi (fix bug). Đôi lúc, khi xây dựng Spider sẽ có thể gặp các lỗi, vì vậy chúng ta nên sử dụng một số phương pháp để debug xem Spider đang gặp vấn đề ở đoạn lệnh vào và sửa lại. Có 4 phương pháp Debug chính với Spider:

* **Parse Command**: Cho phép kiểm tra hành vi của các phần khác nhau của Spider ở cấp hàm (method level). Nó có ưu điểm là linh hoạt và dễ sử dụng, nhưng không cho phép gỡ lỗi mã bên trong một method.
* **Scrapy Shell**: Mặc dù Parse Command rất hữu ích để kiểm tra hành vi của một Spider, nhưng việc kiểm tra những gì xảy ra bên trong một call\_back sẽ giúp ích rất ít cho việc hiển thị phản hồi đã nhận và kết quả đầu ra. Scrapy Shell sẽ giúp gỡ lỗi trong những tình huống như vậy.
* **Open in browser**: Sử dụng hàm open\_in\_browser để xem một response trông như thế nào trong trình duyệt.
* **Logging**: Logging là một tùy chọn hữu ích khác để nhận thông tin về lần chạy Spider. Mặc dù không thuận tiện nhưng nó có lợi thế là các log sẽ có sẵn trong tất cả các lần chạy trong tương lai nếu chúng cần thiết.

### Splash

Splash cung cấp một công cụ để hiển thị mã javaScript cho khung trình thu thập thông tin Scrapy. Nó có các chức năng sau:

* Trả về HTML của một trang web.
* Xử lý đồng thời nhiều trang.
* Tắt việc tải hình ảnh của trang web, tăng tốc độ xử lý.
* Thực thi các đoạn code JS do người dùng định nghĩa.
* Được viết ở dạng Lua Script.

Như vậy Splash sẽ có thể giải quyết được các vấn đề khi chúng ta cần tương tác với trang web mới có thể lấy được dữ liệu.

Graphical user interface, text

Description automatically generated

*Hình 8 Giao diện Splash*

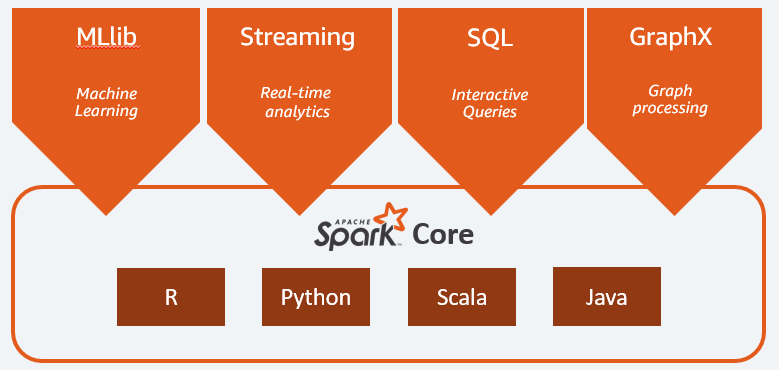
## Apache Spark

### Giới thiệu Apache Spark

**Apache Spark** là một framework mã nguồn mở, cho phép bạn xây dựng những mô hình dự đoán nhanh chóng với khả năng thực hiện tính toán cùng lúc trên một nhóm các máy tính hay trên toàn bộ các tập dữ liệu mà không cần thiết phải trích xuất các mẫu tính toán thử nghiệm. Tốc độ xử lý dữ liệu của Apache spark có được là do khả năng thực hiện các tính toán trên nhiều máy khác nhau cùng một lúc tại bộ nhớ trong (in-memories) hay hoàn toàn trên RAM.

Spark cũng rất dễ dàng để sử dụng do có thể được lập trình bằng Python, Java hoặc Scala. Spark sẽ gồm các thành phần chính như sau:

* **Spark Core**: Đây được xem là nền tảng và điều kiện cho sự vận hành của các thành phần còn lại của Apache Spark, thành phần này đảm nhận vai trò thực hiện các công việc tính toán, xử lý trong bộ nhớ và tham chiếu các dữ liệu được lưu trữ tại các hệ thống lưu trữ ở bên ngoài.
* **Spark SQL**: Thành phần này giúp thực hiện các thao tác trên các Dataframe bằng các ngôn ngữ như Java, Scala hay Python thông qua sự hỗ trợ của ngôn ngữ SQL. Đồng thời cũng cung cấp SchemaRDD với mục đích hỗ trợ cho các kiểu structured data và semi-structured data.
* **Spark Streaming**: Mục đích sử dụng của thành phần này chính là coi stream là các mini batches và thực hiện các kỹ thuật RDD transformation với các dữ liệu này để phân tích stream. Điều này giúp việc xử lý stream và phát triển lambda architecture trở nên dễ dàng bằng cách tận dụng lại các đoạn code được viết để xử lý batch.
* **MLlib**: Là một nền tảng học máy, Spark MLlib nhanh hơn gấp 9 lần so với phiên bản chạy trên Hadoop (theo so sánh của benchmark) nhờ kiến trúc phân tán dựa trên bộ nhớ.
* **GraphX**: Đây là nền tảng xử lý các đồ thị dựa trên Spark. Nó cung cấp các API và được sử dụng để diễn tả tất cả các tính toán có trong đồ thị.



*Hình 10 Các thành phần của Spark*

### Giới thiệu về SparkSQL

SparkSQL là một trong những thành phần chính của Spark. DataFrame là một API bậc cao hơn RDD (Resilient Distributed Dataset) được Spark giới thiệu vào năm 2013 (từ Apache Spark 1.3). Tương tự như RDD, dữ liệu trong DataFrame cũng được quản lý theo kiểu phân tán và không thể thay đổi (immutable distributed). Tuy nhiên dữ liệu này được sắp sếp theo các cột, tương tự như trong các cơ sở dữ liệu quan hệ. DataFrame được phát triển để giúp người dùng có thể dễ dàng thực hiện các thao tác xử lý dữ liệu cũng như làm tăng đáng kể hiệu quả xử lý của hệ thống.

Khi hoạt động, các Data Processing Engine như Spark sẽ cần đọc dữ liệu từ một Data Source nào đó, và cũng sẽ cần lưu các dữ liệu sau khi được xử lý vào một Data Slink nào đó. Các nguồn dữ liệu này có thể được chia ra làm hai loại.

**External**: Đây là các nguồn dữ liệu nằm ngoài hệ thống Big Data của bạn, ví dụ như một Database từ hệ thống khác, hoặc là các file dữ liệu từ hệ thống. Một số loại External Data Source như sau:

* JDBC Data Source: SQL Server, Oracle, PostgreSQL,...
* NoSQL Data System: MongoDB, Cassandra,...
* Cloud Data Warehouse.
* Stream Integrators: Kafka,...

**Internal**: Ngược lại với External, đây là các nguồn dữ liệu nằm trực tiếp bên trong hệ thống Big Data như HDFS, AWS S3,... Và để lấy dữ liệu thì bạn sẽ truy cập trực tiếp vào dữ liệu.

### Đọc/Ghi dữ liệu

Để đọc/ghi dữ liệu từ các file, cần cung cấp một số thông tin cơ bản để thực hiện việc đọc dữ liệu như:

* **Format**: Loại file cần đọc/ghi.
* **Path**: Đường dẫn đến file cần đọc/ghi.
* **Mode**: Cơ chế đọc file, do DataFrame làm việc với các dữ liệu có cấu trúc và các cơ chế đọc file sẽ có từng cách xử lý khác nhau. Các mode đọc file:
* *Permissive*: Tất cả các trường được đặt thành null và các bản ghi bị hỏng được đặt trong một cột được gọi là \_corrupt\_record.
* *DropMalformed*: Xóa các hàng bị lỗi.
* *FailFast*: Đưa ra lỗi cho hệ thống khi có hàng bị lỗi.
* **Schema**: Cung cấp cấu trúc của dữ liệu muốn đọc hoặc có thể để hệ thống tự thiết lập các Schema đó nếu như dữ liệu có cấu trúc đơn giản.

### Sử dụng Spark SQL

Để có thể thực hiện được các câu lệnh SQL, cơ chế này sẽ gồm 4 bước như sau và sẽ được thực hiện thông qua Spark SQL Engine:

* **Analysic**: Spark sẽ đọc và tạo nên một Abstract Syntax Tree, từ có sẽ lấy được các thông tin như tên bảng, tên cột,...
* **Logical Optimization**: Spark sẽ tối ưu hóa lại các truy vấn để tiết kiệm về thời gian thực thi.
* **Physical Planning**: Spark sẽ chọn ra phương án tối ưu nhất để thực thi.
* **Code Generation**: Chuyển hóa truy vấn về các câu lệnh Java rồi thực hiện.

Trong Spark cũng có Database trong đó, vậy nên bạn có thể tạo các Database và từ đó tạo các Table và View ở trong chính Spark. Table ở trong Spark sẽ có hai phần:

* **Table Data**: Được lưu trữ dưới dạng file ở trong hệ thống lưu trữ phân tán.
* **Table Metedata**: Hay còn được gọi là catalog, lưu các thông tin về bảng và dữ liệu của bảng đó như Schema, tên bảng, phân vùng,... Các dữ liệu này có thể được lưu theo Spark Session hoặc ở Hive.

Spark Table cũng được chia thành hai loại:

* **Managed Tables**: Dữ liệu ở Table sẽ được lưu ở một đường dẫn là *spark.sql.warehouse.dir* (do Cluster Admin quy định). Và khi bạn xóa Table này đi thì Spark cũng sẽ xóa cả Metadata và dữ liệu của bảng.
* **Unmanaged Tables**: Khi tạo bảng này thì bạn sẽ cần khai báo thêm cả về nơi mà Table Data sẽ được lưu trữ (có thể là một Data Store ở bên ngoài), lúc này Spark sẽ chỉ tạo Table Metadata để lưu trữ các thông tin về bảng. Khi bạn xóa Table này thì Spark cũng chỉ xóa Metadata chứ không xóa dữ liệu.

### Row và Column

Một số thao tác phổ biến như:

* Hợp nhất các Dataframe (JOIN, UNION).
* Tổng hợp lại dữ liệu từ nhiều Dataframe (joining, windowing, rollups)
* Sử dụng các hàm và các Tranformation có sẵn.
* Sử dụng các hàm tự định nghĩa.

Do Dataframe sẽ lưu trữ các dữ liệu có cấu trúc theo các cột và hàng. Vậy nên các thao tác cũng sẽ được chia ra để sử dụng cho các cột và hàng. Các thao tác với hàng sẽ biến đổi từng hàng trong Dataframe thành một hàng mới, ví dụ như sau:

Graphical user interface, application

Description automatically generated

*Hình 12 Data Transformation trong Spark*

### User Defined Function

Ngoài sử dụng các hàm có sẵn trong Spark, chúng ta cũng có thể tự định nghĩa để xử lý dữ liệu được gọi là **UDF (User Defined Function)**. Cách này sẽ thuận tiện hơn khi thực hiện các xử lý phức tạp. UDF cũng sẽ hỗ trợ hai cách thức là qua String Expression hoặc Object Expression, với mỗi cách thức sử dụng này thì việc đăng ký và sử dụng UDF cũng sẽ khác nhau. Để sử dụng được cơ chế này cần 3 bước:

* Khai báo hàm bạn sẽ sử dụng để xử lý dữ liệu.
* Đăng ký hàm đó thành một UDF.
* Sử dụng hàm.

### Data Aggregation

Join là thao tác vô cùng quan trọng nếu như bạn muốn tổng hợp dữ liệu từ nhiều Dataframe khác nhau. Do Spark hỗ trợ xử lý dữ liệu phân tán vậy nên có thể dữ liệu trong Dataframe sẽ ở các phân vùng khác nhau. Từ đó dẫn đến một vấn đề khi JOIN đó là có thể bị thiếu dữ liệu, do dữ liệu tương ứng đang ở một phân vùng khác.

Table

Description automatically generated

*Hình 13 Minh họa Spark join*

Lúc này Spark sẽ sử dụng một cơ chế gọi là Internal Joins and Shuffle, cơ chế này hoạt động khá giống với MapReduce. Ở Stage đầu tiên dữ liệu sẽ được tạo các id dựa trên điều kiện JOIN (join key) sau đó chuyển xuống các node để thực hiện việc JOIN dữ liệu. Điều đặc biệt là dữ liệu được gửi xuống các node sẽ được shuffle lại sao cho các id của cả hai bảng đều trùng nhau giúp tránh việc thiếu sót dữ liệu.

Việc sử dụng Internal Joins and Shuffle sẽ tốn rất nhiều tài nguyên do bạn sẽ cần di chuyển dữ liệu sang các node, phần vùng khác nhau. Vì vậy, đối với các dạng dữ liệu không phù hợp, nên sử dụng phương án chia bucket cho Dataframe, phương án này nên được sử dụng nếu dữ liệu chúng ta cần join nhiều lần, và chắc chắn sẽ cần phải thực hiện JOIN.

## Tableau

### Giới thiệu Tableau

Tableau là giải pháp đến từ Mỹ, phát triển giải pháp business intelligence nhằm xây dựng nền tảng số & phân tích dữ liệu cho doanh nghiệp

Trong thời gian 7 năm trở lại đây, Tableau luôn là giải pháp đứng đầu trong các bình chọn đánh giá cho giải pháp business intelligence như Gartner,..

Với mục tiêu Nhận diện & Hiểu dữ liệu vận hành của doanh nghiệp bạn, dựa trên cơ sở Dễ sử dụng & Đẹp đã mang lại cho Tableau trở thành công cụ luôn được các nhà quản trì hàng đầu trên thế giới sử dụng để vận hành doanh nghiệp mình.

### Tính năng chính của Tableau

* Kết nối với nhiều nguồn dữ liệu của nhiều hệ quản trị dữ liệu khác nhau, và làm việc đồng thời trên trên các nguồn dữ liệu này.
* Cung cấp công cụ để bạn làm sạch và chuẩn bị dữ liệu cho việc phân tích
* Dễ dàng tạo các phân tích dữ liệu bằng các kéo thả đơn giản.
* Cung cấp các công thức thường gặp khi xử lý số liệu như so sánh cùng kỳ, kỳ trước đó, chênh lệch,...và bạn dễ dàng tạo ra các công thức cho riêng mình từ các chiều hoặc công thức hiện tại
* Tạo các tương tác lọc, rút trích, drill-up, drill-down hoặc các tham số xử lý dữ liệu ngay trên các biểu đồ
* Phân tích dữ liệu trên bản đồ hành chính với bản đồ Việt Nam và Quốc tế theo từng tỉnh thành ở Việt Nam và chi tiết trên từng đường phố của bản đồ
* Giúp cho người dùng nghiệp vụ dễ dàng tạo ra các dashboard quản trị và các story của dữ liệu vận hành doanh nghiệp
* Vận hành tốt trên các thiết bị di động iOS & Android giúp cho bạn luôn luôn có thông tin, dữ liệu và phân tích dữ liệu mọi lúc mọi nơi
* Chia sẻ các phân tích dữ liệu với đồng nghiệp
* Linh động chọn môi trường triển khai với on-premise, on-cloud với full cloud trên nền tảng của Tableau hoặc các public cloud của Amazon, Google hoặc Microsoft. Hỗ trợ các hệ điều hành Windows, Linux, OS,...

1. **PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**
   1. **Một số mô hình use case**

Diagram

Description automatically generated

*Hình 3.1 Mô hình UseCase Tổng quát.*

Diagram

Description automatically generated

*Hình 3.2 Sequence Diagram Đăng nhập*

Diagram

Description automatically generated

*Hình 3.3 Statechart Diagram Thêm sản phẩm*

* 1. **Các bảng trong cơ sở dữ liệu**
     1. **Bảng Manufacturer**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Kích thước (Bytes)** | **Chú thích** | **Ràng buộc** |
| 1 | id | varchar(20) | 20 | Id nhà sản xuất | Khóa chính |
| 2 | text | varchar(100) | 100 | Tên nhà sản xuất |  |

* + 1. **Bảng Series**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Kích thước (Bytes)** | **Chú thích** | **Ràng buộc** |
| 1 | id | varchar(100) | 20 | Id phiên bản | Khóa chính |
| 2 | text | varchar(100) | 100 | Tên phiên bản |  |
| 3 | img | varchar(100) | 100 | Hình ảnh chi tiết phiên bản |  |
| 4 | id\_img | varchar(100) | 100 | Hình ảnh demo |  |
| 5 | manu\_id | varchar(20) | 20 | Id nhà sản xuất | Khóa ngoại liên kết với bảng Manu(id) |

* + 1. **Bảng Product**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Kích thước (Bytes)** | **Chú thích** | **Ràng buộc** |
| 1 | id | int | 4 | Id sản phẩm | Khóa chính |
| 2 | url | varchar(100) | 100 | Link tới hình sản phẩm |  |
| 3 | name | varchar(100) | 100 | Tên sản phẩm |  |
| 4 | img | Varchar(100) | 100 | Hình ảnh sản phẩm |  |
| 5 | price | Int | 4 | Giá sản phẩm |  |
| 6 | article | Varchar(1000) | 1000 | Đoạn văn miêu tả |  |
| 7 | screen | Varchar(100) | 100 | Màn hình |  |
| 8 | os | Varchar(100) | 100 | Hệ điều hành |  |
| 9 | camera\_truoc | Varchar(100) | 100 | Camera trước |  |
| 10 | camera\_sau | Varchar(100) | 100 | Camera sau |  |
| 11 | cpu | Varchar(100) | 100 | Chip |  |
| 12 | ram | Varchar(100) | 100 | Ram |  |
| 13 | rom | Varchar(100) | 100 | Rom |  |
| 14 | microsd | Varchar(100) | 100 | Microsd |  |
| 15 | pin | Varchar(100) | 100 | Pin |  |
| 16 | series\_id | Varchar(100) | 100 | Id phiên bản | Khóa ngoại liên kết tới bảng Version(id) |

* + 1. **Bảng Color**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Kích thước (Bytes)** | **Chú thích** | **Ràng buộc** |
| 1 | id | Varchar(20) | 20 | Id màu | Khóa chính |
| 2 | text | Varchar(100) | 100 | Tên màu |  |
| 3 | Img\_demo | Varchar(100) | 100 | Hình ảnh demo màu sắc |  |
| 4 | modal\_id | int | 4 | Màu ứng với cấu hình | Khóa ngoại liên kết với bảng Manu(id) |

* + 1. **Bảng Customer**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Kích thước (Bytes)** | **Chú thích** | **Ràng buộc** |
| 1 | id | char(7) | 7 | Id khách hàng | Khóa chính |
| 2 | password | nvarchar(20) | 20 | Mật khẩu |  |
| 3 | fullname | nvarchar(30) | 30 | Họ tên |  |
| 4 | phone | char(10) | 10 | Điện thoại |  |
| 5 | address | nvarchar(200) | 200 | Địa chỉ |  |

* + 1. **Bảng Comment**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Kích thước (Bytes)** | **Chú thích** | **Ràng buộc** |
| 1 | customer\_id | varchar(20) | 20 | Mã khách hàng | Khóa chính |
| 2 | product\_id | int | 4 | Mã sản phẩm |  |
| 3 | time\_up | timestamp | 6 | Thời gian đăng |  |
| 4 | content | varchar(3000) | 3000 | Nội dung |  |
| 5 | rate-star | int | 4 | Đánh giá |  |
| 6 | time\_resolved | timestamp | 6 | Thời gian giải quyết |  |
| 7 | handler | varchar(20) | 20 | Bộ phận giải quyết |  |

* + 1. **Bảng Branch**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Kích thước (Bytes)** | **Chú thích** | **Ràng buộc** |
| 1 | id | int | 4 | Mã chi nhánh | Khóa chính |
| 2 | address | varchar(140) | 140 | Địa chỉ |  |
| 3 | district | varchar(60) | 60 | Quận/Huyện |  |
| 4 | province | varchar(20) | 20 | Tỉnh/Thành phố |  |
| 5 | longitude | float | 8 | Kinh độ |  |
| 6 | latitude | float | 8 | Vĩ độ |  |

* + 1. **Bảng Invoice**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Kích thước (Bytes)** | **Chú thích** | **Ràng buộc** |
| 1 | id | int | 4 | Mã hóa đơn | Khóa chính |
| 2 | customer\_id | varchar(50) | 50 | Mã khách hàng | Khóa ngoại |
| 3 | status | int | 4 | Trạng thái |  |
| 4 | time\_  invoice | timestamp | 6 | Thời gian đặt |  |
| 5 | time\_  receive | timestamp | 6 | Thời gian nhận |  |
| 6 | branch\_id | int | 4 | Chi nhánh | Khóa ngoại |
| 7 | address\_  receive | varchar(200) | 200 | Địa chỉ nhận |  |
| 8 | receiver\_  name | varchar(100) | 100 | Tên người nhận |  |
| 9 | receiver\_  phone | char(10) | 10 | Điện thoại người nhận |  |
| 10 | note | varchar(1000) | 1000 | Ghi chú |  |

* + 1. **Bảng Invoice\_detail**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Kích thước (Bytes)** | **Chú thích** | **Ràng buộc** |
| 1 | invoice\_  id | int | 4 | Mã hóa đơn | Khóa ngoại/chính |
| 2 | product\_  id | int | 4 | Mã sản phẩm | Khóa ngoại/chính |
| 3 | color\_id | int | 4 | Mã màu sắc | Khóa ngoại/chính |
| 4 | quantity | int | 4 | Số lượng |  |
| 5 | product\_price | int | 4 | Giá đơn vị |  |
| 6 | discount | int | 4 | Giảm giá |  |

* 1. **Lược đồ cơ sở dữ liệu**

Diagram

Description automatically generated

*Hình 3.4 Lược đồ cơ sở dữ liệu quan hệ quản lý bán điện thoại di động*

1. **THỰC NGHIỆM**
   1. **Mô tả hệ thống**

Hệ thống sử dụng PostgreSQL làm cơ sở dữ liệu chính, React JS làm giao diện frontend, NodeJS thực hiện việc liên kết giữa cơ sở dữ liệu và giao diện trang web.

Đối với hệ thống thu thập, tải và trực quan hóa dữ liệu thì sử dụng python làm ngôn ngữ chính, spark làm công cụ xử lý và tải dữ liệu thô vào cơ sở dữ liệu, Tableau để trực quan hóa dữ liệu.

* 1. **Các phần mềm yêu cầu**
* ReactJS
* NodeJS
* Docker (có PostgreSQL – password: 12345, Splash 3)
* Python 3 trở lên.
* Tableau (có thể sử dụng tài khoản email \*.edu để đăng ký miễn phí)
  1. **Giao diện phần mềm**

Graphical user interface, application, website

Description automatically generated

*Hình 4.1 Giao diện trang danh sách sản phẩm*

Graphical user interface, application

Description automatically generated

*Hình 4.2 Giao diện trang chi tiết sản phẩm*

Graphical user interface, application, website

Description automatically generated

*Hình 4.3 Giao diện trang đăng nhập*

Graphical user interface, application, website

Description automatically generated

*Hình 4.4 Giao diện trang đăng ký*

* 1. **Code thiết kế**

Text

Description automatically generated

*Hình 4.5 Code thiết kế Cart Reducer*

Graphical user interface, text, application

Description automatically generated

*Hình 4.6 Code thiết kế Phone Detail Reducer*

Graphical user interface, text, application

Description automatically generated

*Hình 4.7 Code kết nối Backend 1*

Graphical user interface, text

Description automatically generated

*Hình 4.8 Code kết nối Backend 2*

KẾT LUẬN

**Kết quả đạt được**

* Xây dựng được cơ sở dữ liệu dựa theo yêu cầu của công ty hoặc khách hàng.
* Hoàn thiện được hệ thống thu thập dữ liệu dựa trên các bài thực hành.
* Tiếp cận được văn hóa, môi trường làm việc tại doanh nghiệp thực tế.

**Hạn chế của bản thân**

* Còn nhiều kiến thức chưa tiếp cận được.
* Kiến thức chuyên ngành chưa sâu, cần hoàn thiện hơn.

TÀI LIỆU THAM KHẢO

[1] Cursor method, <https://docs.mongodb.com/manual/reference/method/js-cursor/>, tham khảo ngày: 6/1/2022.

[2] int, bigint, smallint, and tinyint (Transact-SQL), [http://msdn.microsoft.com/en-us/library/ms187745.aspx/](https://getbootstrap.com/docs/4.5/getting-started/introduction/), tham khảo ngày: 6/1/2022.

[3] Install SQL Server Integration Services in Visual Studio 2019, <https://www.mssqltips.com/sqlservertip/6481/install-sql-server-integration-services-in-visual-studio-2019/>, tham khảo ngày: 6/1/2022.

[4] Regex Cheatsheet, https://www.dataquest.io/blog/regex-cheatsheet/, tham khảo ngày: 6/1/2022.

[5] Selector, <https://doc.scrapy.org/en/latest/topics/selectors.html>, tham khảo ngày: 6/1/2022.