

ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC SÀI GÒN



NHÓM 10

Triệu Khánh Quang

BÁO CÁO CÁ NHÂN

ĐỒ ÁN MÔN

CÁC CÔNG NGHỆ LẬP TRÌNH HIỆN ĐẠI

NGÀNH: CÔNG NGHỆ THÔNG TIN

Giảng viên phụ trách

ThS. PHẠM THI VƯƠNG

Thành phố Hồ Chí Minh, năm 2024

BÁO CÁO CÁ NHÂN**Họ tên:** Triệu Khánh Quang**MSSV:** 3120410428**Tên đề án:** Tìm hiểu về Appsheet – Quản lý nhà kho**Nhóm:** 10**Học phần:** Các công nghệ lập trình hiện đại**Mã:** 841072

STT	Công việc đã làm
1	Ví dụ về nền tảng no-code phổ biến
2	Tổng quan về ưu điểm của Appsheet
3	Tổng quan về nhược điểm của Appsheet
4	Nghiên cứu cách kết nối dữ liệu trên Appsheet
5	Nghiên cứu cách định dạng dữ liệu trên Appsheet
6	Nghiên cứu tạo Views trên Appsheet
7	Nghiên cứu backup dữ liệu

Chi tiết các công việc đã làm**Ví dụ về nền tảng no-code phổ biến**

1.4. Ví dụ về nền tảng no-code phổ biến:

- AppSheet: Nền tảng no-code mạnh mẽ cho phép kết nối với nhiều nguồn dữ liệu và tạo ứng dụng di động đa nền tảng.
- Adalo: Nền tảng no-code thân thiện với người dùng, cung cấp các công cụ trực quan để xây dựng ứng dụng di động đẹp mắt.
- Bubble: Nền tảng no-code linh hoạt, cho phép tạo ứng dụng web và di động phức tạp với nhiều tính năng.

6

- Glide: Nền tảng no-code tập trung vào việc tạo ứng dụng từ bảng tính Google Sheets, Airtable và các nguồn dữ liệu khác.

Các nền tảng App No-Code là một công cụ mạnh mẽ và hữu ích giúp mọi người dễ dàng tạo ứng dụng mà không cần kiến thức lập trình chuyên sâu. Với sự phát triển không

Nội dung từ trang 5 đến trang 6

Tổng quan về ưu điểm của Appsheet

1.5.1. Ưu điểm

- Dễ sử dụng: Giao diện kéo thả trực quan, dễ dàng sử dụng cho mọi người, kể cả những người không có kinh nghiệm lập trình.
- Nhanh chóng: Tạo ứng dụng nhanh chóng chỉ trong vài phút thay vì vài tháng.
- Tiết kiệm chi phí: Giảm thiểu chi phí phát triển ứng dụng đáng kể so với phương pháp truyền thống.
- Mạnh mẽ: Tạo ứng dụng với đầy đủ chức năng và khả năng tùy chỉnh cao.
- Kết nối dữ liệu: Kết nối với nhiều nguồn cơ sở dữ liệu khác nhau như Google Sheet, Google Drive, Salesforce, SQL [Server](#),...
- Đa dạng ứng dụng: Tạo ứng dụng cho mọi nhu cầu, từ quản lý công việc, kinh doanh đến theo dõi sức khỏe, giáo dục,...
- Miễn phí cho các ứng dụng cá nhân: AppSheet có sẵn miễn phí cho các ứng dụng cá nhân.

Nội dung trong trang 7

Tổng quan về nhược điểm của Appsheet

1.5.2. Nhược điểm

- So với các nền tảng phát triển ứng dụng truyền thống, AppSheet vẫn còn một số thiếu sót như:
 - Xử lý dữ liệu phức tạp.
 - Tích hợp với các hệ thống bên ngoài.
 - Quản lý người dùng và quyền truy cập.
 - Tùy chỉnh giao diện và chức năng nâng cao hơn.
- Khả năng mở rộng: với các ứng dụng quy mô lớn, khả năng mở rộng dữ liệu, chức năng và hiệu suất có thể bị giới hạn.
- Khả năng tùy chỉnh:

8

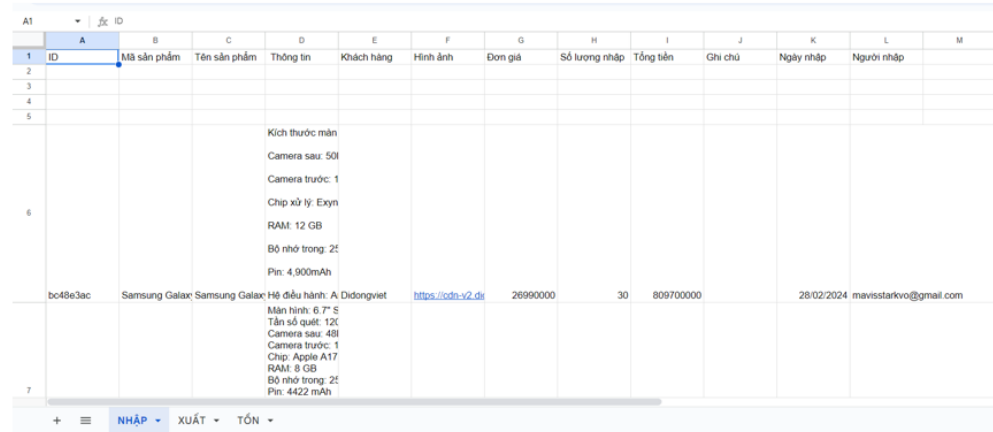
- Mặc dù có khả năng tùy chỉnh cao, nhưng AppSheet vẫn có thể hạn chế trong việc tạo giao diện và chức năng độc đáo.
- Giao diện kéo thả có thể không đáp ứng được nhu cầu thiết kế phức tạp.
- Hỗ trợ: Hỗ trợ khách hàng của AppSheet có thể hạn chế so với các nền tảng trả phí.

Nội dung trong trang 7 trang 8

Nghiên cứu cách kết nối dữ liệu trên Appsheet

2.2.1. Kết nối dữ liệu

Dữ liệu là nền tảng cho mọi ứng dụng Appsheet. Để có một ứng dụng Appsheet hoàn chỉnh chúng ta cần phải đầu tư khá nhiều cho dữ liệu của mình. Appsheet cũng sẽ cung cấp các công cụ để kết nối với nhiều nền tảng lưu trữ dữ liệu khác nhau: Google Sheets, Airtable, Salesforce, SQL Server, [mySQL](#),....



ID	Mã sản phẩm	Tên sản phẩm	Thông tin	Khách hàng	Hình ảnh	Đơn giá	Số lượng nhập	Tổng tiền	Ghi chú	Ngày nhập	Người nhập
bc48e3ac	Samsung Galax	Samsung Galax	Hệ điều hành: A Didongviet Kích thước màn Camera sau: 50l Camera trước: 1 Chip xử lý: Exyn RAM: 12 GB Bộ nhớ trong: 256 Pin: 4.900mAh		https://cdn.v2.dz	26990000	30	809700000		28/02/2024	mavisstarkvo@gmail.com
			Màn hình: 6.7" S Tần số quét: 120 Camera sau: 48l Camera trước: 1 Chip: Apple A17 RAM: 8 GB Bộ nhớ trong: 256 Pin: 4422 mAh								

HÌNH 2.1. DỮ LIỆU ĐƯỢC LƯU TRỮ BẰNG GOOGLE SHEETS

Nội dung từ trang 14 đến trang 15

Nghiên cứu cách định dạng dữ liệu trên Appsheet

2.2.2. Định dạng dữ liệu

Một cơ sở dữ liệu cơ bản sẽ bao gồm các hàng và cột. Trong đó cái tên thường phản ánh nội dung và bản chất của đối tượng hoặc biến mà nó được dùng để đặt tên. Đặt tên sao cho nó có ý nghĩa đúng với nội dung đã khó mà đặt tên sao cho đúng với quy tắc chuẩn cũng là điều khó không kém. Đặt tên theo quy tắc chuẩn sẽ giúp những người khác khi xem code của bạn cũng cảm thấy dễ hiểu hơn vì nó một quy tắc đã được định sẵn

- Sử dụng một kí tự alphabet ở đầu tên, không sử dụng các kí tự " _ " gạch dưới, hoặc số ở đầu tên biến.
- Giới hạn việc viết tắt quá nhiều trong tên biến hoặc đối tượng. Điều này dễ gây nhầm lẫn.
- Giới hạn việc sử dụng các từ viết tắt hai lần trong tên.
- Đặt tên mang ý nghĩa đúng với nội dung và tính chất của biến hoặc đối tượng.
- Không sử dụng khoảng trắng(space) hoặc các kí tự đặc biệt trong tên như *,\$,#...
- Đặt tên khóa chính nên có hậu tố Id thường thì công thức sẽ là tên bảng + "Id".
- Đối với khóa ngoại thì nên có tên giống với khóa chính mà nó tham chiếu tới. Ví dụ khóa chính của bảng Staff là "StaffId" và bảng [Class] có khóa ngoại đến bảng Staff thì nó có tên là "StaffId".
- Hạn chế sử dụng tiếng Việt có dấu khi đặt mặc dù Appsheet không cấm.

Nội dung từ trang 15 đến trang 16

Nghiên cứu tạo Views trên Appsheet

2.3. View

Views (Chế độ xem) là một trong những tính năng quan trọng nhất của Appsheet, cho phép bạn hiển thị và tương tác với dữ liệu trong ứng dụng của mình theo nhiều cách khác nhau. Appsheet cung cấp nhiều loại views khác nhau để đáp ứng mọi nhu cầu, bao gồm:

- Dashboard View: Hiển thị tổng quan về dữ liệu của bạn với các biểu đồ, bảng, chỉ số và thẻ.
- Table View: Hiển thị dữ liệu dạng bảng với khả năng sắp xếp, lọc và tìm kiếm.
- Form View: Tạo biểu mẫu để thu thập dữ liệu từ người dùng.
- Detail View: Hiển thị chi tiết cụ thể về một mục dữ liệu.
- Card View: Hiển thị dữ liệu dạng thẻ với hình ảnh, tiêu đề và mô tả.
- Gallery View: Hiển thị hình ảnh, video hoặc các tệp tin khác trong thư viện.
- Map View: Hiển thị dữ liệu trên bản đồ.
- Deck View: Hiển thị dữ liệu dạng thẻ có thể lật sang để xem thêm thông tin.
- Onboarding View: Giới thiệu ứng dụng và hướng dẫn người dùng sử dụng.

Khi thao tác trên views ta cần lưu ý những điều như sau:

- Trên ứng dụng, tên view nên được đặt tên theo tên bảng (table) mà view đó sẽ hiển thị. Ví dụ: nếu ta tạo view cho table SANPHAM, thì tên table sẽ là SANPHAM hoặc SANPHAMUSER. Phần display name có thể được đặt theo tùy thích.

Nội dung từ trang 16 đến trang 18

Nghiên cứu backup dữ liệu

4.5. Backup dữ liệu

Appsheets cho phép chúng ta tạo ứng dụng một cách dễ dàng nhưng vẫn có một nhược điểm là khi chúng ta ghi nhận một lượng dữ liệu lớn thì appsheet đồng bộ dữ liệu khá là chậm, mất rất nhiều thời gian để đồng bộ được các dữ liệu về cho chúng ta, như thế sẽ ảnh hưởng đến trải nghiệm của người dùng. Để khắc phục vấn đề này thì đến định kì nào đó thì chúng ta tiến hành sao lưu dữ liệu (Backup dữ liệu).

Quy trình backup dữ liệu như sau:

- Đầu tiên sẽ copy ứng dụng mà chúng ta cần thực hiện backup sau đó lưu lại. (Vd: dữ liệu này nằm ở năm 2024 thì sẽ copy ra sau đó đặt tên ứng dụng là QLK_2024)
- Sau đó tiến hành xoá hết dữ liệu trên ứng dụng gốc mà chúng ta đang sử dụng. (Vd: dữ liệu chủ yếu nằm ở mục PhieuNX, PhieuNXChitiet thì chỉ cần xoá dữ liệu ở 2 trang đó mà thôi).
- Trích xuất dữ liệu từ các bảng mà ta mong muốn sau đó đưa dữ liệu trở lại trên ứng dụng mới.

Nội dung trong trang 85