**ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**NHÓM 10**

**TÌM HIỂU VỀ APPSHEET – QUẢN LÝ NHÀ KHO**

**ĐỀ CƯƠNG ĐỒ ÁN MÔN**

**CÁC CÔNG NGHỆ LẬP TRÌNH HIỆN ĐẠI**

NGÀNH: CÔNG NGHỆ THÔNG TIN

**Thành phố Hồ Chí Minh, năm 2024**

**ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**NHÓM 10**

**TÌM HIỂU VỀ APPSHEET – QUẢN LÝ NHÀ KHO**

**ĐỀ CƯƠNG 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**

# **Lời cam đoan**

Em tên là Võ Đăng Quang – người viết báo cáo này, em đại diện nhóm 10 xin cam đoan rằng đồ án “*Tìm hiểu về Appsheet – Quản lý nhà kho*” là công trình nghiên cứu của nhóm 10 dưới sự hướng dẫn trực tiếp của ThS. Phạm Thi Vương.

Tất cả số liệu, thông tin, kết quả nghiên cứu, mọi trích dẫn sử dụng trong báo cáo đều được thu thập và thực hiện một cách trung thực, khách quan, ghi rõ nguồn tài liệu tham khảo theo đúng quy định.

Nhóm em xin hoàn toàn chịu trách nhiệm về nội dung và chịu mọi hình thức kỷ luật theo quy định nếu có bất kì hành vi vi phạm, gian trá nào.

**Thành phố Hồ Chí Minh, năm 2024**

**Nhóm 10**

# **Lời cảm ơn**

Lời đầu tiên nhóm em xin bày tỏ lời cảm ơn chân thành và sâu sắc nhất đến thầy ThS. Phạm Thi Vương, người trực tiếp hướng dẫn, góp ý và tận tình chỉ bảo cho em cho tới khi nhóm hoàn thành đồ án của mình.

Tiếp đến nhóm xin giành lời cảm ơn đến quý thầy cô Trường Đại học Sài Gòn – khoa Công nghệ thông tin đã truyền đạt cho em những kiến thức vô cùng quý báu và bổ ích trong suốt quá trình nghiên cứu và học tập tại trường.

Và cuối cùng, chân thành cảm ơn tới những thành viên trong nhóm đã luôn sát cánh cùng nhau, những lời động viên, những lần hỗ trợ những lúc cần thiết đã phần nào giúp cả nhóm hoàn thành đồ án này.

# **Tổng hợp**

Nhóm: 10

Tên đề tài: Tìm hiểu về Appsheet – Quản lý nhà kho

Thành viên nhóm:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **MSSV** | **Họ và Tên** | **Khối lượng công việc** |
| 1 | 3120410429 | Võ Đăng Quang | 25% |
| 2 | 3120410437 | Nguyễn Văn Tấn Quân | 25% |
| 3 | 3120410428 | Triệu Khánh Quang | 25% |
| 4 | 3120410450 | Nguyễn Quốc Siêu | 25% |

Kế hoạch:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tuần** | **Thời gian** | **Công việc thực hiện** | **Ghi chú** |
| 1 | 15/1/2024 – 21/1/2024 | Thành lập nhóm 10 | Trao đổi thông qua Zalo |
| 2 | 22/1/2024 – 28/1/2024 | * Nhận gợi ý đề tài từ giảng viên hướng dẫn. * Nghiên cứu tên đề tài. |  |
| 3 | 19/2/2024 – 25/2/2024 | * Thống nhất đề tài. * Tìm hiêu các nền tảng khác có liên quan đến đề tài. * Nộp đề cương chi tiết. | Trao đổi thông qua Zalo, Discord và Github |
| 4 | 26/2/2024 – 3/3/2024 |
| 5 | 4/3/2024 – 10/3/2024 | * Nghiên cứu các chức năng của Appsheet * Trao đổi với giảng viên các vấn đề khó khăn gặp phải | Các vấn đề đã được giải quyết sau khi có được sự gợi ý từ giảng viên |
| 6 | 11/3/2024 – 17/3/2024 |
| 7 | 18/3/2024 – 24/3/2024 | * Nghiên cứu các chức năng cơ bản của Appsheet * Hoàn tất chương 1 | Bắt đầu viết báo cáo |
| 8 | 25/3/2024 – 31/3/2024 | * Khởi động dự án * Hoàn tất chương 2 và 3 |  |
| 9 | 1/4/2024 – 7/4/2024 |  |
| 10 | 8/4/2024 – 14/4/2024 |  |
| 11 | 15/4/2024 – 21/4/2024 |  |
| 12 | 22/4/2024 – 28/4/2024 | * Kiểm thử và kết thúc đồ án | Hoàn tất báo cáo |
| 13 | 29/4/2024 – 5/5/2024 | Báo cáo đồ án theo lịch mà giảng viên hướng dẫn đã đưa ra trước đó |  |
| 14 | 6/5/2024 – 12/5/2024 |  |
| 15 | 13/5/2024 – 19/5/2024 |  |

# **Mục lục**

[**Lời cam đoan** i](#_Toc164422837)

[**Lời cảm ơn** ii](#_Toc164422838)

[**Tổng hợp** iii](#_Toc164422839)

[**Mục lục** v](#_Toc164422840)

[**Danh mục hình ảnh** vi](#_Toc164422841)

[**Danh mục bảng biểu** vi](#_Toc164422842)

[**Lời mở đầu** 1](#_Toc164422843)

[**Chương 1: TỔNG QUAN VỀ APPSHEET** 3](#_Toc164422844)

[**1.1.** **Tổng quan về Appsheet** 3](#_Toc164422845)

[**1.1.1.** **Ưu điểm** 3](#_Toc164422846)

[**1.1.2.** **Nhược điểm** 4](#_Toc164422847)

[**1.2.** **Cách Appsheet hoạt động** 5](#_Toc164422848)

[**1.3.** **Sử dụng Appsheet trong doanh nghiệp** 5](#_Toc164422849)

[**Chương 2: NGHIÊN CỨU VỀ APPSHEET** 7](#_Toc164422850)

[**2.1.** **Giới hạn của Appsheet** 7](#_Toc164422851)

[**2.2.** **Dữ liệu và cách làm việc với dữ liệu trên Appsheet** 8](#_Toc164422852)

[**2.2.1.** **Kết nối dữ liệu** 8](#_Toc164422853)

[**2.2.2.** **Định dạng dữ liệu** 9](#_Toc164422854)

[**2.3.** **View** 10](#_Toc164422855)

[**2.4.** **Action** 12](#_Toc164422856)

[**2.5.** **Automation** 14](#_Toc164422857)

[**2.6.** **Các chức năng nâng cao trong Appsheet** 15](#_Toc164422858)

[**2.6.1.** **Tính năng nâng cao về dữ liệu:** 15](#_Toc164422859)

[**2.6.2.** **Tính năng nâng cao về ứng dụng:** 16](#_Toc164422860)

[**2.6.3.** **Tính năng quản lý và cộng tác:** 16](#_Toc164422861)

[**Chương 3: ỨNG DỤNG APPSHEET ĐỂ XÂY DỰNG PHẦN MỀM QUẢN LÝ NHÀ KHO** 17](#_Toc164422862)

[**Kết luận và hướng phát triển** 18](#_Toc164422863)

[**Tài liệu tham thảo** 19](#_Toc164422864)

# **Danh mục hình ảnh**

[Hình 1.1. Một số nền tảng tạo ứng dụng không cần code phổ biến hiện nay 1](#_Toc164718347)

[Hình 1.2. App là gì 3](#_Toc164718348)

[Hình 1.3. Một số lợi ích của App No-code so với App truyền thống 4](#_Toc164718349)

[Hình 1.4. Praveen Seshadri (trái) và Brian Sabino (phải), đồng sáng lập Appsheet 6](#_Toc164718350)

[Hình 1.5. Cách Appsheet hoạt động 9](#_Toc164718351)

[Hình 1.6. AppSheet trong doanh nghiệp 10](#_Toc164718352)

[Hình 2.1. Dữ liệu được lưu trữ bằng Google Sheets 14](#_Toc164718353)

[Hình 2.2. Tên views 17](#_Toc164718354)

[Hình 2.3. View options 17](#_Toc164718355)

[Hình 2.4. Icon minus đối với các views không cần cho người nhìn thấy 17](#_Toc164718356)

[Hình 2.5. Icon hiển thị đồng nhất 18](#_Toc164718357)

[Hình 2.6. Sử dụng màu trong view 18](#_Toc164718358)

[Hình 2.7. Đánh số thứ tự cho các action 20](#_Toc164718359)

**No table of figures entries found.**

# **Danh mục bảng biểu**

[Bảng 1.1. Các doanh nghiệp ba mức quy mô sử dụng Appsheet 11](#_Toc164718332)

[Bảng 2.1. Bảng so sánh giới hạn của 2 phiên bản appsheet 14](#_Toc164718339)

# **Lời mở đầu**

Xu hướng tạo ứng dụng di động và web không cần code đang ngày càng phát triển mạnh mẽ. Nhu cầu xây dựng ứng dụng ngày càng cao, tuy nhiên không phải ai cũng có khả năng lập trình. Do đó, các nền tảng tạo ứng dụng không cần code ra đời như một giải pháp hữu hiệu.





*Hình 1.1. Một số nền tảng tạo ứng dụng không cần code phổ biến hiện nay*

Hiện nay có rất nhiều nền tảng tạo ứng dụng không cần code phổ biến như: AppSheet, Zoho Creator, Airtable, Google App Maker, Microsoft Power Apps, Mendix, OutSystems,... Mỗi nền tảng có ưu và nhược điểm riêng, phù hợp với nhu cầu sử dụng khác nhau. Trong đồ án này nhóm em chọn AppSheet làm nền tảng nghiên cứu chính.

AppSheet là một nền tảng cho phép bạn xây dựng các ứng dụng di động và web tùy chỉnh mà không cần viết mã. Nó sử dụng giao diện kéo, thả đơn giản để bạn có thể tạo các ứng dụng dựa trên dữ liệu trong Google Sheets, Google Drive và các nguồn dữ liệu khác có sẵn miễn phí cho các ứng dụng cá nhân. Ngoài ra, AppSheet còn có một số ưu điểm khác so với các nền tảng khác:

* Nhanh chóng: AppSheet cho phép bạn tạo ứng dụng nhanh chóng hơn so với các nền tảng khác.
* Dễ học: AppSheet dễ học hơn so với các nền tảng khác.
* Hỗ trợ tốt: AppSheet có cộng đồng hỗ trợ mạnh mẽ và đội ngũ hỗ trợ khách hàng chuyên nghiệp.

Những nghiên cứu về tính hiệu quả của Appsheet sẽ được giải thích vì sao các doanh nghiệp áp dụng nó cho hệ thống của mình. Vì vậy, để tìm hiểu rõ hơn về cách hoạt động của Appsheet cũng như cách các doanh nghiệp sẽ ứng dụng nó, nhóm em quyết định chọn nghiên cứu đề tài: “Tìm hiểu về Appsheet – Quản lý nhà kho”.

Trong đề tài này, nhóm sẽ trình bày rõ thế nào là một phần mềm được tạo từ Appsheet. Kết hợp với việc quan sát trực quan trên các video hướng dẫn và các bài tập đã làm, nhóm sẽ cho ra các nhận định của mình. Và cuối cùng, nhóm em sử dụng phương pháp thực nghiệm để xây dựng một phần mềm bằng Appsheet.

Đồ án của nhóm 10 được chia thành các phần như sau:

Chương 1: Trình bày tổng quan về khái niệm Appsheet. Đồng thời, nhóm em cũng trình bày rõ ưu và nhược điểm của nó.

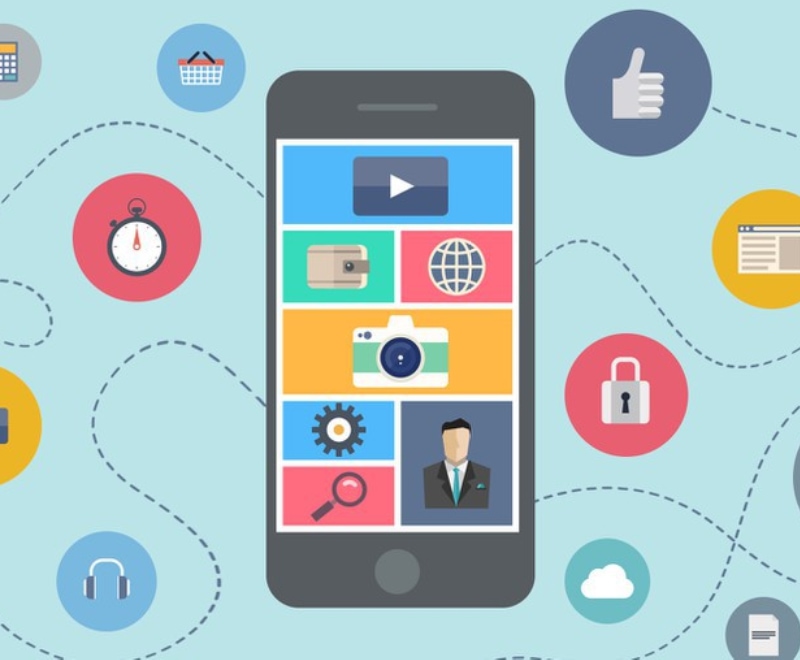
Chương 2: Trong chương này, nhóm sẽ trình bày chi tiết hơn về các chức năng của Appsheet từ cơ bản đến nâng cao mà nhóm đã nghiên cứu được. Bên cạnh đó cũng sẽ trình bày các chức năng của các nền tảng có liên quan sẽ được sử dụng.

Chương 3: Xây dựng phần mềm từ khâu đặc tả cho tới chạy thử và đưa ra kết luận cũng như hướng phát triển cho phần mềm.

# **Chương 1: TỔNG QUAN VỀ APPSHEET**

## **Giới thiệu về App No-code**

Trước khi tìm hiểu App No-code là gì ta phải hiểu app là gì trước đã. Nói một cách đơn giản App (Ứng dụng) là một chương trình phần mềm được thiết kế để thực hiện các tác vụ cụ thể trên thiết bị điện tử như smartphone, tablet, laptop,…



*Hình 1.2. App là gì*

App có thể được cài đặt từ các kho ứng dụng như App Store (iOS), Google Play Store (Android), Microsoft Store (Windows) hoặc tải trực tiếp từ trang web của nhà phát triển.

Một App có thể có nhiều mục đích sử dụng khác nhau. Và để phát triển các ứng dụng, ta phải code. Để làm được điều đó ta cần các lâp trình viên tham gia quá trình xây dựng, phát triển. Một App xuất hiện trên các kho ứng dụng cần rất nhiều thời gian, công sức và tiền bạc để phát triển. Vậy thì phát triển một App No-code nghĩa là như thế nào?

Nền tảng App No-Code (hay còn gọi là nền tảng phát triển ứng dụng không cần viết mã) là một hệ thống cho phép người dùng tạo ứng dụng di động, web và máy tính mà không cần có kiến thức lập trình chuyên sâu. Thay vì viết mã truyền thống, người dùng sử dụng các công cụ trực quan như kéo thả, chọn lựa và cấu hình để xây dựng giao diện, logic và chức năng cho ứng dụng.

## **Lợi ích của việc sử dụng App No-code**



*Hình 1.3. Một số lợi ích của App No-code so với App truyền thống*

* Tiết kiệm thời gian và chi phí: Việc phát triển ứng dụng no-code diễn ra nhanh chóng hơn nhiều so với lập trình truyền thống, giúp giảm thiểu chi phí cho nhân sự và nguồn lực.
* Dễ sử dụng: Giao diện trực quan và các công cụ đơn giản giúp mọi người, bất kể trình độ kỹ thuật, đều có thể tạo ứng dụng.
* Tính linh hoạt: Nền tảng No-code cung cấp nhiều tính năng và khả năng tùy chỉnh, đáp ứng nhu cầu đa dạng của người dùng.
* Khả năng mở rộng: App No-code có thể dễ dàng mở rộng và nâng cấp để đáp ứng nhu cầu phát triển trong tương lai.

## **Ứng dụng trong công việc hiệu quả**

### **Dưới góc nhìn của lập trình viên**

* Phải có code.
* Phải làm chủ nền tảng.
* Dữ liệu phải lưu local.
* Phải nắm được source của phần mềm.

### **Dưới góc nhìn của doanh nghiệp**

* Mọi phần mềm, app chỉ đơn giản là công cụ.
* Đối với doanh nghiệp để chọn một nền tàng thích hợp cần phân tích nhu cầu, ngân sách và hiệu quả mang lại. Các tiêu chí được ưu tiên như sau:
* Hiệu quả
* Bảo mật
* Hỗ trợ lâu dài
* Chi phí hợp lý
* Giải quyết được vấn đề cho doanh nghiệp
* Dễ dàng kết nối với các hệ sinh thái trong tương lai

Suy cho cùng, No-code hay code cũng chỉ là công cụ. Mục đích cuối cùng là tối ưu hoá công việc. Để áp dụng App No-code vào công việc hiệu quả, chúng ta cần nắm vững các yếu tố sau:

* Nắm rõ mục tiêu chính trong quá trình tạo App, loại bỏ các phần gây xao nhãng.
* Biết và hiểu về nền tảng, điểm mạnh, điểm hạn chế.
* Xử lý các vấn đề tồn đọng hiện tại của doanh nghiệp.
* Chi phí phù hợp với yêu cầu doanh nghiệp.
* Định hướng phát triển trong tương lai.

## **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.
* 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 ngừng của công nghệ no-code, ngày càng nhiều người sẽ có thể tiếp cận và sử dụng nền tảng này để hiện thực hóa ý tưởng sáng tạo của mình.

## **Tổng quan về Appsheet**

Appsheet được phát triển bởi công ty AppSheet Inc. được thành lập vào năm 2014 bởi Praveen Seshadri và Brian Sabino.



*Hình 1.4. Praveen Seshadri (trái) và Brian Sabino (phải), đồng sáng lập Appsheet*

Năm 2020, Appsheet được mua lại bởi Google và trở thành một phần của Google Cloud Platform.

Đây là nền tảng tự động hoá giúp bạn tạo mobile app hay web một cách nhanh chóng mà không cần code. Nền tảng này đơn giản, trực quan và dễ sử dụng, cho phép chúng ta dễ dàng biến ý tưởng thành hiện thực chỉ với vài thao tác đơn giản. Với AppSheet, không cần có kinh nghiệm lập trình để tạo và quản lý các ứng dụng. Để sử dụng AppSheet, chỉ cần có kiến thức cơ bản về bảng tính hoặc cơ sở dữ liệu và có thể sử dụng các công cụ kéo và thả để tạo ra ứng dụng hoàn chỉnh. Tuy nhiên, ta nên cân nhắc nhu cầu của mình và biết những hạn chế của AppSheet trước khi quyết định sử dụng nó.

### **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 cở 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.

### **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:
* 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. Thời gian chờ đợi hỗ trợ có thể lâu hơn và tài nguyên hỗ trợ có thể ít hơn.

Tuy nhiên, AppSheet là một nền tảng vừa được phổ biến gần đây nên không thể tránh khỏi việc có các nhược điểm nêu trên. Nhờ cộng đồng ngày một tăng và liên tục được cải tiến, cập nhật thêm tính năng mới, Appsheet sẽ trở thành một công cụ đắc lực cho doanh nghiệp trong tương lai.

## **Cách Appsheet hoạt động**

AppSheet là một nền tảng đột phá cho phép bạn tạo ứng dụng di động và web mạnh mẽ mà không cần viết mã. Nền tảng sử dụng giao diện kéo thả trực quan, giúp bạn dễ dàng biến ý tưởng thành hiện thực chỉ với vài thao tác đơn giản.

Với AppSheet, bạn có thể kết nối với nhiều nguồn dữ liệu khác nhau như Google Trang tính, Google Drive, Salesforce, SQL Server, v.v. và tạo ứng dụng cho mọi nhu cầu, từ quản lý công việc, kinh doanh đến theo dõi sức khỏe, giáo dục,…

### **Dữ liệu**

Biểu tượng bảng tính đại diện cho nguồn dữ liệu của bạn, chẳng hạn như Google Trang tính, Google Drive, Salesforce, SQL Server, v.v. AppSheet kết nối với nguồn dữ liệu của bạn và truy xuất dữ liệu để sử dụng trong ứng dụng.

### **Giao diện**

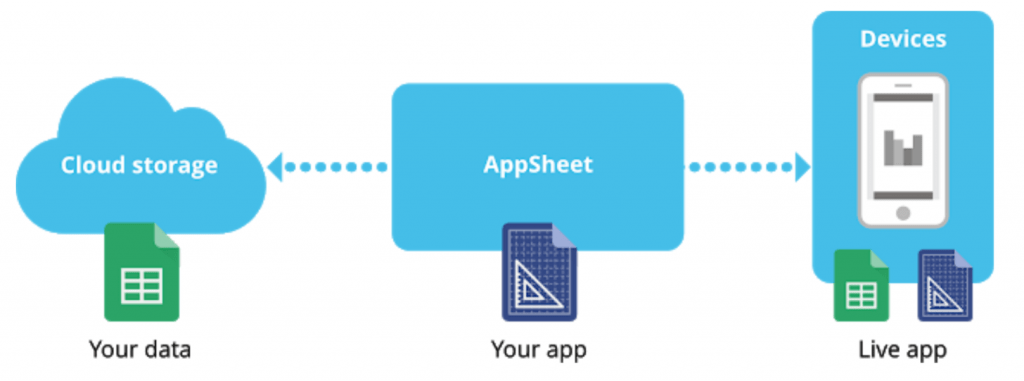
Biểu tượng điện thoại và máy tính đại diện cho giao diện người dùng của ứng dụng di động và web. Bạn sử dụng giao diện kéo thả trực quan để thiết kế giao diện người dùng cho ứng dụng của mình.

### **Chức năng**

Biểu tượng bánh răng đại diện cho logic có hướng dữ liệu được sử dụng để xác định chức năng cho ứng dụng của bạn. AppSheet cung cấp các công cụ để tạo quy tắc, xử lý sự kiện và tự động hóa các tác vụ.

### **Triển khai**

Biểu tượng AppSheet Marketplace đại diện cho khả năng chia sẻ ứng dụng của bạn với người dùng khác hoặc công bố lên AppSheet Marketplace. Bạn có thể quản lý và cập nhật ứng dụng của bạn sau khi triển khai.



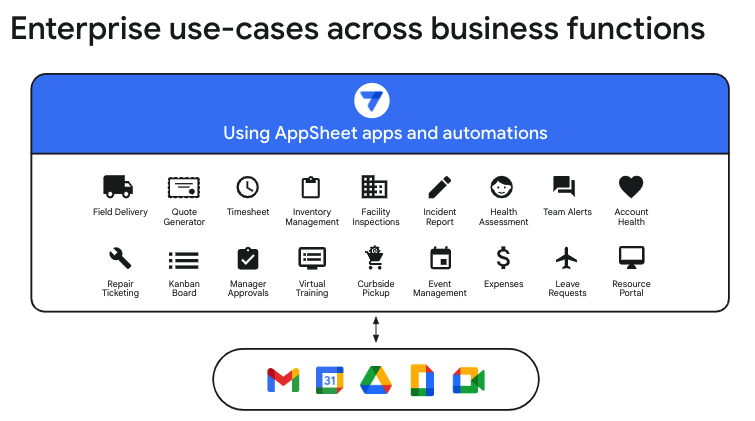
*Hình 1.5. Cách Appsheet hoạt động*

Trong hình 1.3, Dữ liệu được lưu trữ trong đám mây, sau đó AppSheet xử lý để tạo ra ứng dụng và cuối cùng, ứng dụng có thể được sử dụng trên các thiết bị di động và web. Hình ảnh cũng cho thấy các bước chính trong quá trình tạo ứng dụng với AppSheet:

* Kết nối dữ liệu: Kết nối AppSheet với nguồn dữ liệu của bạn.
* Thiết kế giao diện: Sử dụng giao diện kéo thả để thiết kế giao diện người dùng cho ứng dụng của bạn.
* Xác định chức năng: Sử dụng logic có hướng dữ liệu để xác định chức năng cho ứng dụng của bạn.
* Kiểm tra và triển khai: Kiểm tra ứng dụng của bạn và triển khai cho người dùng.

Nhìn chung, hình ảnh cung cấp một cái nhìn tổng quan về cách thức hoạt động của AppSheet và các bước chính trong quá trình tạo ứng dụng.

## **Sử dụng Appsheet trong doanh nghiệp**



*Hình 1.6. AppSheet trong doanh nghiệp*

AppSheet là một nền tảng mạnh mẽ, với nó, người dùng có thể thu thập dữ liệu phong phú thông qua biểu mẫu, mã vạch, địa điểm, chữ ký, và hình ảnh, đồng thời tùy chỉnh các tính năng, trải nghiệm người dùng, bảo mật,… Nền tảng này cũng cung cấp khả năng quản lý người dùng với bảo mật dựa trên vai trò, xác thực, và báo cáo sử dụng.

AppSheet phù hợp với mọi quy mô doanh nghiệp. Với các doanh nghiệp nhỏ, tạo ứng dụng đơn giản để quản lý vừa tiết kiệm được chi phí thuê kĩ thuật viên lập trình lại vừa tăng hiệu quả hoạt động. Còn đối với doanh nghiệp vừa và lớn, để tối ưu hóa hiệu quả và mở rộng thị trường thì ta cần thiết kế một ứng dụng phức tạp hơn cho các quy trình kinh doanh chuyên biệt nhưng vẫn đảm bảo về mặt chi phí. Đây cũng không phải điều bất khả thi kể cả khi Appsheet còn nhiều hạn chế khi thiết kế một ứng dụng phức tạp nên các doanh nghiệp vừa và lớn vẫn cứ yên tâm sử dụng Appsheet.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Quy mô doanh nghiệp | Nhu cầu | Ứng dụng | Lợi ích | Ví dụ |
| Nhỏ | Quy trình đơn giản, nguồn lực hạn chế | Quản lý dự án, theo dõi khách hàng tiềm năng, quản lý chi tiêu | Tiết kiệm chi phí, tăng hiệu quả | Quán cà phê: Quản lý đơn hàng, theo dõi nguyên liệu, thu ngân  Cửa hàng bán lẻ: Quản lý kho hàng, theo dõi doanh thu, cập nhật giá cả |
| Vừa | Quy trình phức tạp hơn, nhiều bộ phận tham gia | Quản lý quy trình bán hàng, quản lý quan hệ khách hàng (CRM), quản lý sản xuất | Tăng năng suất, cải thiện dịch vụ khách hàng, tối ưu hóa quy trình | Doanh nghiệp bán lẻ: Quản lý đơn hàng trực tuyến, theo dõi hàng tồn kho, hỗ trợ khách hàng trực tuyến  Công ty dịch vụ: Quản lý lịch trình nhân viên, theo dõi tiến độ công việc, báo cáo khách hàng |
| Lớn | Quy trình phức tạp, nhiều dữ liệu, nhiều người dùng | Tích hợp hệ thống ERP, tự động hóa quy trình, phân tích dữ liệu | Tăng hiệu quả hoạt động, giảm rủi ro, đưa ra quyết định sáng suốt | Doanh nghiệp sản xuất: Quản lý sản xuất, theo dõi chất lượng sản phẩm, giám sát dây chuyền sản xuất  Công ty tài chính: Phân tích dữ liệu tài chính, quản lý rủi ro, báo cáo cho ban lãnh đạo |

*Bảng 1.1. Các doanh nghiệp ba mức quy mô sử dụng Appsheet*

AppSheet giúp tự động hóa các quy trình làm việc, từ việc gửi và tự động hóa email, SMS, và thông báo đẩy. Điều này giúp doanh nghiệp tập trung vào công việc quan trọng hơn và giảm thiểu công việc thủ công. Ngoài ra, AppSheet có khả năng tích hợp mạnh mẽ với Google Workspace và các ứng dụng của bên thứ ba như Office 365, Dropbox, và Salesforce.com, cũng như với các cơ sở dữ liệu SQL, Apigee, REST APIs,… Điều này cho phép doanh nghiệp kết nối dữ liệu và phần mềm vào một nền tảng thống nhất, tạo điều kiện cho việc triển khai và quản lý ứng dụng một cách hiệu quả.

Với AppSheet, các doanh nghiệp nhỏ cũng có thể cạnh tranh với các doanh nghiệp lớn bằng cách phát triển và triển khai các ứng dụng tùy chỉnh nhanh chóng để đáp ứng nhu cầu cụ thể của họ. Điều này làm cho AppSheet trở thành một công cụ quan trọng trong việc thúc đẩy sự đổi mới và tối ưu hóa quy trình làm việc trong doanh nghiệp.

# **Chương 2: NGHIÊN CỨU VỀ APPSHEET**

## **Giới hạn của Appsheet**

Appsheet là một nền tảng đa năng để tạo ứng dụng di động và web mà không cần phải code. Tuy nhiên, nó cũng có một số giới hạn nhất định giữa phiên bản miễn phí và trả phí mà ta cần lưu ý:

|  |  |  |
| --- | --- | --- |
|  | Phiên bản miễn phí | Phiên bản trả phí |
| Giới hạn dữ liệu | * Giới hạn 10000 hàng cho mỗi bảng dữ liệu. * Giới hạn 20 bảng cho mỗi cơ sở dữ liệu. * Giới hạn 20 cơ sở dữ liệu cho mỗi người dùng. | Loại bỏ các giới hạn của phiên bản miễn phí và cung cấp lượng lưu trữ dữ liệu lớn hơn. |
| Giới hạn tính năng | * Cung cấp các tính năng cơ bản để tạo ứng dụng đơn giản. * Một số tính năng nâng cao bị giới hạn hoặc không khả dụng. | Mở khóa tất cả các tính năng, bao gồm logic nâng cao, tự động hóa, tích hợp API và hơn thế nữa. |
| Giới hạn hiệu suất | Hiệu suất có thể bị ảnh hưởng bởi độ phức tạp của ứng dụng, lưu trữ và cấu trúc kết nối dữ liệu. Tối ưu hóa truy vấn dữ liệu và chia nhỏ các bảng dữ liệu là giải pháp mà Appsheet đưa ra để cải thiện hiệu suất. | |
| Giới hạn tuỳ chỉnh | Khả năng tùy chỉnh giao diện người dùng của ứng dụng bị giới hạn ở một mức độ nào đó. | |
| Hỗ trợ | Phiên bản miễn phí chỉ cung cấp hỗ trợ cộng đồng qua diễn đàn và tài liệu. | Phiên bản trả phí cung cấp hỗ trợ qua email và số điện thoại. |
| Chia sẻ người dùng | Một mail được chia sẻ miễn phí cho 9 mail dùng tất cả các app. | Tăng giới hạn mail có thể chia sẻ |

*Bảng 2.1. Bảng so sánh giới hạn của 2 phiên bản appsheet*

Các gói trả phí:

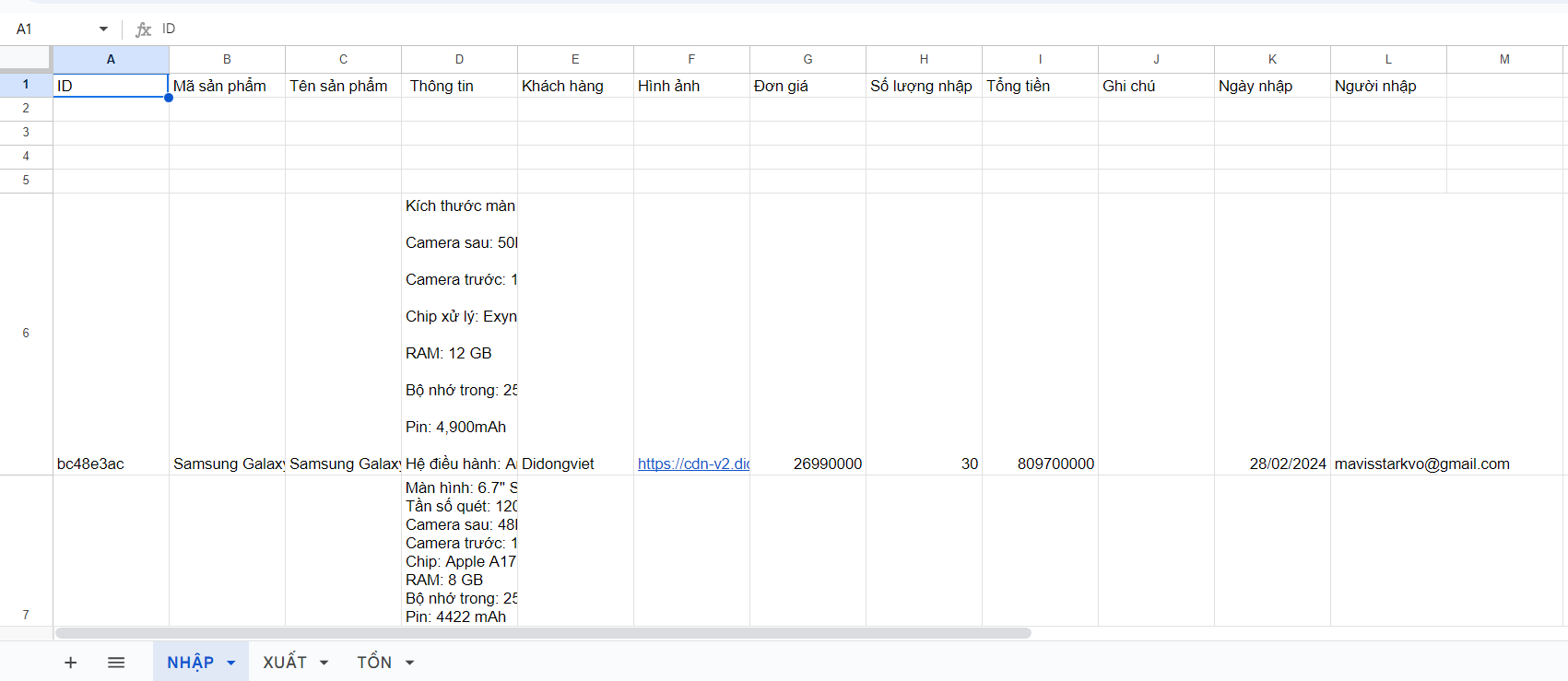
* Stater (Loại thường): 5$/mail/tháng
* Core (Loại trung bình): 10$/mail/tháng
* Public (Loại dành cho app công khai): 50$/app/tháng
* Enterprise (Loại dành cho doanh nghiệp): 25-35$/mail/tháng

Theo như nghiên cứu được các gói nâng cấp càng có giá cao thì sẽ cho người dùng nhiều tính năng hữu ích. Ví dụ như gói Stater có tính năng Quick Sync hay gói Core có tính năng lọc dữ liệu nâng cao cực kì tối ưu cho người dùng.

## **Dữ liệu và cách làm việc với dữ liệu trên Appsheet**

### **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,….



*Hình 2.1. Dữ liệu được lưu trữ bằng Google Sheets*

Sau khi kết nối dữ liệu, Appsheet sẽ cung cấp các tính năng sau để làm việc với dữ liệu như:

* Tạo các biểu mẫu và bảng: Hiển thị dữ liệu của bạn ở dạng dễ đọc và dễ sử dụng.
* Lọc và sắp xếp dữ liệu: Tìm kiếm thông tin cụ thể một cách nhanh chóng và hiệu quả.
* Thực hiện các phép tính: Tính toán tổng, trung bình, v.v. trên dữ liệu của bạn.
* Tạo biểu đồ và đồ thị: Hình dung dữ liệu của bạn để dễ dàng hiểu hơn.
* Tự động hóa quy trình làm việc: Tự động hóa các tác vụ lặp đi lặp lại để tiết kiệm thời gian và công sức.
* Gửi email và thông báo: Nhận thông báo khi có thay đổi dữ liệu quan trọng.

Trong đề tài này, nhóm em dùng Google Sheets làm bảng dữ liệu vì nó phù hợp với quy mô ứng dụng vừa và nhỏ.

### **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 chê sử dụng tiếng Việt có dấu khi đặt mặc dù Appsheet không cấm.

Khi tuân thủ các quy tắc trên ta sẽ dễ dàng thay đổi nền tảng cơ sở dữ liệu khi cần mà không cần tốn công sức để chỉnh sửa tên các cột, hàng.

Trong thực tế, khi thiết kế một ứng dụng, ta không thể tránh được việc người dùng nhập dữ liệu không theo ý mình. Ví dụ ta muốn người dùng nhập 04/05/2024 theo định dạng là dd/MM/YYYY thay vì 4/5/24 hoặc 5/4/2024. Việc nhập liệu không đúng định dạng như này sẽ ảnh hưởng rất nhiều về mặt thời gian cho người lập trình. Sau khi có bảng dữ liệu, việc đầu tiên ta cần làm là phải định dạng các kiểu dữ liệu cho từng cột. Việc này có thể mất đôi chút thời gian những kết quả đạt được lại rất hiệu quả về sau.

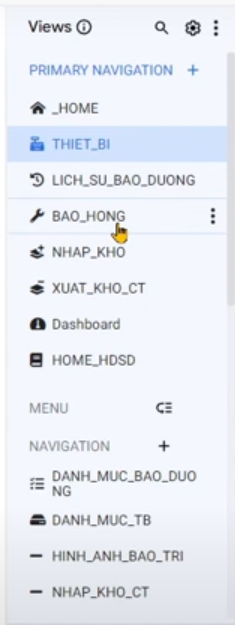
## **View**

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.

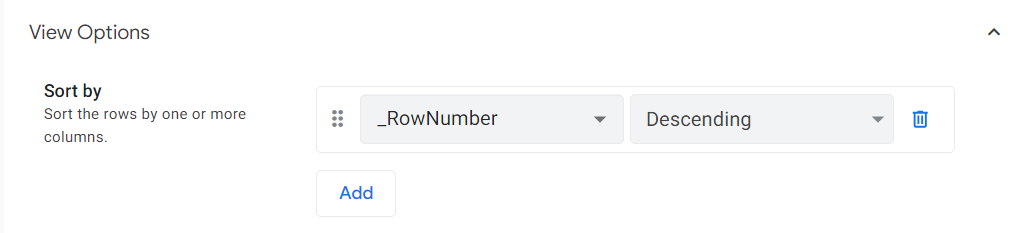
Khi thao tác trên views ta cần lưu ý những điều như sau:

* 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 tuỳ thích.



*Hình 2.2. Tên views*

Ở đây chúng ta chọn phần views trên thanh công cụ để xem tất cả những view hiện có. Trong phần hiển thị này cho thấy những phần view nào đang hiển thị cho table nào. Điều đó giúp cho quá trình làm việc trở nên dễ dàng hơn rất nhiều.



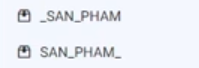
*Hình 2.3. View options*

* Tiếp đến trong phần view options ta nên thiết lập sort by là \_Rownumber và chọn decending để hiển thị các dòng mới nhất lên đầu tiên.



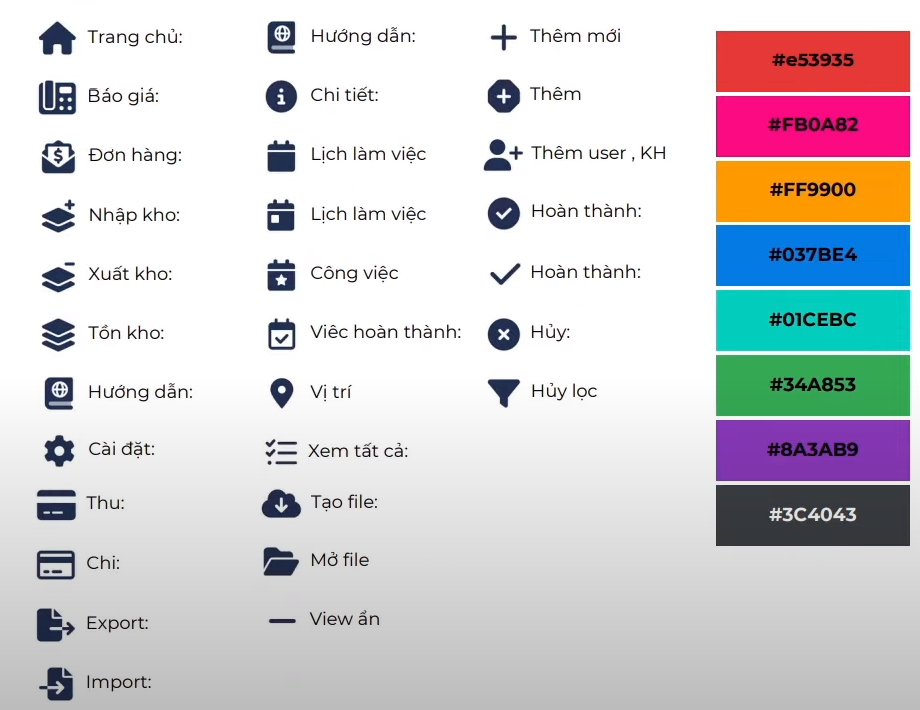
*Hình 2.4. Icon minus đối với các views không cần cho người nhìn thấy*

* Icon cho views ẩn nên được hiển thị với icon minus



*Hình 2.5. Icon hiển thị đồng nhất*

* Đồng nhất icon: các icon nên có hình ảnh đồng nhất với nhau. Ví dụ: vuông, tròn hoặc hình ảnh tương tự nhau như hình 2.5.



*Hình 2.6. Sử dụng màu trong view*

* Cuối cùng là màu sắc trên app: Nên sử dụng tối đa 5 màu trên một view để người dùng dễ nhìn và không bị nhầm lẫn.

## **Action**

Actions là một tính năng mạnh mẽ trong Appsheet cho phép bạn tự động hóa các tác vụ, mở rộng chức năng và tạo ra các ứng dụng phức tạp hơn. Actions có thể được sử dụng để thực hiện nhiều loại nhiệm vụ, bao gồm:

* Điều hướng giữa các views: Di chuyển người dùng đến các màn hình khác trong ứng dụng của bạn.
* Thay đổi dữ liệu: Thêm, xóa, cập nhật dữ liệu trong các bảng.
* Gửi email: Gửi email cho người dùng hoặc các bên liên quan khác.
* Tạo báo cáo: Tạo báo cáo PDF hoặc CSV từ dữ liệu của bạn.
* Gọi API: Kết nối với các API bên ngoài để truy cập dữ liệu và dịch vụ khác.
* Tự động hóa quy trình làm việc: Thực hiện các tác vụ lặp đi lặp lại một cách tự động.

Có bốn loại Actions chính trong Appsheet:

* User Actions: Được kích hoạt bởi người dùng, chẳng hạn như khi họ nhấp vào nút hoặc nhập dữ liệu vào biểu mẫu.
* Data Actions: Được kích hoạt bởi sự thay đổi dữ liệu, chẳng hạn như khi thêm hoặc cập nhật dữ liệu trong bảng.
* Schedule Actions: Được thực hiện tự động theo lịch trình, chẳng hạn như vào một thời điểm cụ thể mỗi ngày hoặc mỗi tuần.
* External Actions: Gọi API bên ngoài để truy cập dữ liệu và dịch vụ khác.

Tạo Actions:

* Chọn loại Action bạn muốn tạo.
* Cung cấp thông tin cần thiết cho Action, chẳng hạn như dữ liệu cần thay đổi, email cần gửi hoặc API cần gọi.
* Thêm logic nếu cần thiết, chẳng hạn như điều kiện hoặc vòng lặp.
* Lưu Actions và thử nghiệm.

Sau khi chúng ta đã tiếp hiểu các lưu ý liên quan đến dữ liệu và hiển thị, bây giờ ta sẽ đến phần lưu ý của action.

* Đối với action bị ẩn: chúng ta sẽ sử dụng biểu tượng minus như hình 2.4



*Hình 2.7. Đánh số thứ tự cho các action*

* Nhóm action: Khi có nhiều hành động được nhóm lại, chúng cần được đánh số theo thứ tự như hình 2.7.
* Đồng bộ hoá app: Các action cần được sắp xếp theo thứ tự cụ thể. Ví dụ: action tạo file cần được đặt trước action mở file. Thứ tự này cần được tuân thủ trong tất cả các bảng của ứng dụng.
* Đối với màu sắc: Nó cũng cần được tuân thủ cho tất cả các hành động. Ví dụ: action duyệt sẽ có màu xanh còn huỷ sẽ có màu đỏ.
* Tên action nên được đặt đồng nhất.

## **Automation**

Automation là một tính năng trong Appsheet cho phép bạn tự động hóa các tác vụ lặp đi lặp lại, tối ưu hóa quy trình làm việc và tiết kiệm thời gian.

Lợi ích của Automation:

* Tiết kiệm thời gian: Tự động hóa các tác vụ lặp đi lặp lại giúp bạn tiết kiệm thời gian và công sức, cho phép bạn tập trung vào những công việc quan trọng hơn.
* Nâng cao hiệu quả: Automation giúp tối ưu hóa quy trình làm việc, loại bỏ sai sót do con người và tăng hiệu quả hoạt động.
* Giảm thiểu lỗi: Automation giúp giảm thiểu sai sót do con người, đảm bảo tính chính xác và nhất quán trong dữ liệu.
* Mở rộng khả năng: Automation giúp bạn tạo ra các ứng dụng phức tạp hơn và mở rộng chức năng của ứng dụng.

Cách sử dụng Automation:

* Xác định tác vụ cần tự động hóa: Xác định các tác vụ lặp đi lặp lại mà bạn muốn tự động hóa.
* Chọn loại Automation: Chọn loại Automation phù hợp với nhu cầu của bạn, chẳng hạn như User Automation, Data Automation, Schedule Automation hoặc External Automation.
* Cung cấp thông tin cần thiết: Cung cấp thông tin cần thiết cho Automation, chẳng hạn như dữ liệu cần thay đổi, email cần gửi hoặc API cần gọi.
* Thêm logic: Thêm logic nếu cần thiết, chẳng hạn như điều kiện hoặc vòng lặp.
* Lưu Automation và thử nghiệm: Lưu Automation và thử nghiệm để đảm bảo nó hoạt động chính xác.

Ví dụ:

* Tự động tạo đơn hàng: Khi khách hàng gửi biểu mẫu đơn hàng, tự động tạo đơn hàng mới trong hệ thống quản lý bán hàng.
* Gửi email thông báo: Khi có đơn hàng mới, tự động gửi email thông báo cho bộ phận kho để chuẩn bị hàng hóa.
* Cập nhật dữ liệu kho: Khi đơn hàng được giao, tự động cập nhật số lượng tồn kho trong hệ thống quản lý kho.
* Gửi báo cáo doanh số: Gửi báo cáo doanh số bán hàng theo ngày, tuần hoặc tháng cho quản lý.

## **Các chức năng nâng cao trong Appsheet**

AppSheet cung cấp nhiều tính năng miễn phí mạnh mẽ để bạn bắt đầu tạo ứng dụng di động và web. Tuy nhiên, để mở rộng khả năng và tạo ra các ứng dụng phức tạp hơn, ta cần nâng cấp lên gói trả phí:

### **Tính năng nâng cao về dữ liệu:**

* Kết nối với nhiều nguồn dữ liệu: Kết nối với nhiều nguồn dữ liệu hơn, bao gồm cơ sở dữ liệu SQL, API RESTful và các tệp tin đám mây.
* Truy vấn dữ liệu phức tạp: Thực hiện truy vấn dữ liệu phức tạp hơn với các bộ lọc, sắp xếp và kết hợp nâng cao.
* Tạo báo cáo và biểu đồ tùy chỉnh: Tạo báo cáo và biểu đồ tùy chỉnh từ dữ liệu của bạn để trực quan hóa thông tin.
* Machine Learning: Tích hợp các mô hình học máy vào ứng dụng của bạn để phân tích dữ liệu và đưa ra dự đoán.

### **Tính năng nâng cao về ứng dụng:**

* Actions nâng cao: Tạo Actions phức tạp hơn với logic tùy chỉnh, kết nối API và xử lý dữ liệu.
* Logic nâng cao: Kiểm soát hành vi ứng dụng bằng cách sử dụng các biểu thức logic phức tạp hơn, vòng lặp và xử lý lỗi.
* Custom Code: Mở rộng khả năng của AppSheet bằng cách thêm mã JavaScript hoặc TypeScript tùy chỉnh để tạo các chức năng và hành vi ứng dụng độc đáo.
* Progressive Web Apps (PWAs): Tạo các ứng dụng web có thể cài đặt và hoạt động như ứng dụng di động với các tính năng nâng cao như truy cập ngoại tuyến và thông báo đẩy.
* Bảo mật nâng cao: Bảo vệ dữ liệu ứng dụng của bạn với các tính năng bảo mật nâng cao như xác thực đa yếu tố, mã hóa dữ liệu tại chỗ và kiểm soát truy cập dựa trên vai trò.

### **Tính năng quản lý và cộng tác:**

* Quản lý người dùng: Quản lý người dùng và quyền truy cập ứng dụng với các tính năng như xác thực người dùng, phân quyền vai trò và nhật ký truy cập.
* Cộng tác nhóm: Cộng tác với các thành viên trong nhóm để phát triển và quản lý ứng dụng với các tính năng như chia sẻ ứng dụng, chỉnh sửa đồng thời và quản lý phiên bản.
* Hỗ trợ nâng cao: Nhận hỗ trợ ưu tiên từ nhóm hỗ trợ AppSheet để giải quyết các vấn đề và nhận trợ giúp với các tính năng nâng cao.

# **Chương 3: ỨNG DỤNG APPSHEET ĐỂ XÂY DỰNG PHẦN MỀM QUẢN LÝ NHÀ KHO**

## **Đặc tả**

## **Tạo dữ liệu và giao diện quản lý thông tin**

## **Chức năng quản lý nhập xuất**

## **Các thiết lập và ràng buộc dữ liệu**

## **Tổng hợp dữ liệu để kiểm kê**

## **Backup dữ liệu**

# **Kết luận và hướng phát triển**

## **Kết luận**

Ông thầy khó ưa

## **Hướng phát triển**

Dí buồi học lại môn này

# **Tài liệu tham thảo**

|  |  |
| --- | --- |
| [1] | AppSheet Creators Community.[online]. <https://www.googlecloudcommunity.com/gc/AppSheet/ct-p/appsheet> |
| [2] | AppSheet Help.[online].  <https://support.google.com/appsheet?hl=en&sjid=3620578480876635593-AP#topic=11981018> |
| [3] | Hoà Data [Học Appsheet Online].[online].  <https://www.youtube.com/@hocdatacunghoa> |
| [4] | Học AppSheet.[online]. <https://www.youtube.com/@hocappsheet> |
| [5] | Nghiện Appsheet.[online]. <https://www.youtube.com/@NghienAppsheet> |