

TRƯỜNG ĐẠI HỌC KINH TẾ - LUẬT ĐHQG -TPHCM
KHOA TÀI CHÍNH – NGÂN HÀNG



ĐỒ ÁN CUỐI KỲ

MÔN HỌC: Gói Phần Mềm Ứng Dụng Cho Tài Chính I

**PHÂN TÍCH BÁO CÁO TÀI CHÍNH TỔNG CÔNG TY CỔ PHẦN
ĐẦU TƯ PHÁT TRIỂN XÂY DỰNG (HOSE: DIG)**

GIẢNG VIÊN HƯỚNG DẪN :	ThS. Ngô Phú Thanh
SINH VIÊN THỰC HIỆN :	Nguyễn Quốc Huy
MÃ HỌC PHẦN :	231CN0801
MSSV :	K214142068

Thành phố Hồ Chí Minh, tháng 12 năm 2023

MỤC LỤC

1.	Lý do chọn mã cổ phiếu DIG:.....	2
2.	Tổng quan về doanh nghiệp:	2
2.1.	Giới thiệu chung	2
2.2.	Lĩnh vực hoạt động.....	2
2.3.	Mục tiêu	3
3.	Tổng quan về khối lượng giao dịch của mã cổ phiếu DIG:	3
4.	Tổng quan về giá giao dịch của mã cổ phiếu DIG:	5
5.	Phân tích kỹ thuật:	6
6.	Phân tích báo cáo tài chính CTCP đầu tư phát triển xây dựng:	10
6.1.	Xem các bảng báo cáo tài chính	11
6.2.	Phân tích bảng cân đối kế toán.....	13
6.3.	Phân tích kết quả hoạt động kinh doanh.....	16
6.4.	Phân tích bảng lưu chuyển tiền tệ.....	19
6.5.	Các chỉ số tài chính	19
6.5.1.	Định giá.....	19
6.5.2.	Khả năng sinh lợi	20
6.5.3.	Sức mạnh tài chính	21
6.5.4.	Hiệu quả quản lý.....	21
6.5.5.	Khả năng hoạt động.....	22
6.6.	So sánh với ngành	22
7.	Tổng kết.....	25
8.	Code	26

1. Lý do chọn mã cổ phiếu DIG:

Lĩnh vực xây dựng và bất động sản là lĩnh vực có tiềm năng tăng trưởng lớn do nhu cầu ngày càng tăng về nhà ở và cơ sở hạ tầng. Cho nên giá trị của các cổ phiếu ngành xây dựng và bất động sản có thể tăng theo thời gian và mang về lợi nhuận đáng kể cho chủ sở hữu. Mã cổ phiếu DIG là một mã cổ phiếu nằm trong lĩnh vực xây dựng và bất động sản. DIG là một trong những nhà phát triển bất động sản sở hữu quỹ đất lớn nhất Việt Nam, với tổng quỹ đất lên tới hơn 8.000 ha, trải dài trên nhiều tỉnh, thành phố, từ Nam ra Bắc. Lọt vào top 10 chủ đầu tư bất động sản uy tín 2022 và lọt vào top 500 doanh nghiệp có lợi nhuận tốt nhất Việt Nam 2023.

Từ tình hình thực tế và sự yêu thích với lĩnh vực xây dựng và bất động sản, em xin quyết định nghiên cứu về mã cổ phiếu này với mong muốn tìm hiểu rõ hơn về mã cổ phiếu DIG, phân tích tình hình tài chính và đánh giá về mã cổ phiếu này.

2. Tổng quan về doanh nghiệp:

2.1. Giới thiệu chung

Công ty Cổ phần Đầu tư Phát triển Xây dựng (mã cổ phiếu: DIG) được thành lập vào ngày 26/05/1990 trực thuộc Bộ xây dựng. Đây là một trong những công ty hàng đầu trong lĩnh vực phát triển bất động sản tại Việt Nam. Tính đến 2022 công ty có tổng tài sản lên tới 14.747,8 tỷ đồng và vốn điều lệ 6.099 tỷ đồng. Và năm 2009, mã cổ phiếu DIG được niêm yết tại Sở Giao dịch Chứng khoán thành phố Hồ Chí Minh.



Hình 1

2.2. Lĩnh vực hoạt động

- Đầu tư kinh doanh, phát triển nhà và hạ tầng kỹ thuật khu đô thị, khu công nghiệp, khu chế xuất, khu công nghệ cao,...
- Tư vấn, môi giới, đấu giá bất động sản, đấu giá quyền sử dụng đất Xây dựng các công trình công nghiệp, dân dụng, kỹ thuật hạ tầng và KCN, công trình giao thông, thủy lợi, cấp, thoát nước
- Lắp đặt các kết cấu thép, các sản phẩm cơ khí và mạ kẽm, các thiết bị cơ điện kỹ thuật công trình ...

- Buôn bán vật liệu, thiết bị lắp đặt khác trong xây dựng.

2.3. Mục tiêu

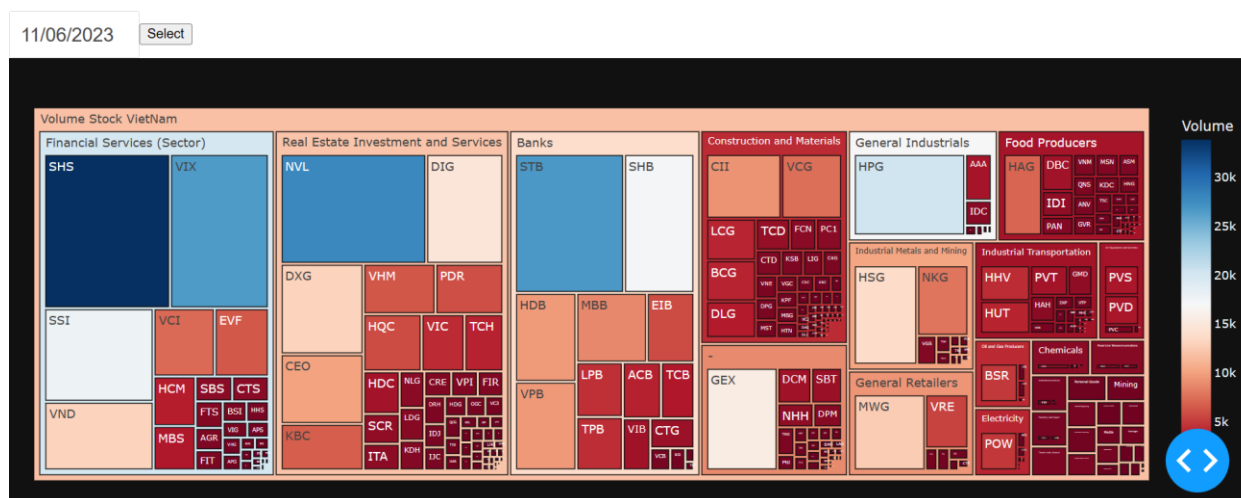
Mục tiêu đến 2025 lọt vào top 5 tập đoàn BĐS lớn nhất Việt Nam

Định hướng phát triển:

- Tập trung đầu tư bất động sản với 3 lĩnh vực:
 - Bất động sản khu đô thị
 - Bất động sản khu du lịch nghỉ dưỡng
 - Bất động sản khu công nghiệp
- Tăng cường hợp tác quốc tế nhằm tạo các công trình kiến trúc hiện đại, đẳng cấp và văn minh.

3. Tổng quan về khối lượng giao dịch của mã cổ phiếu DIG:

Select Date



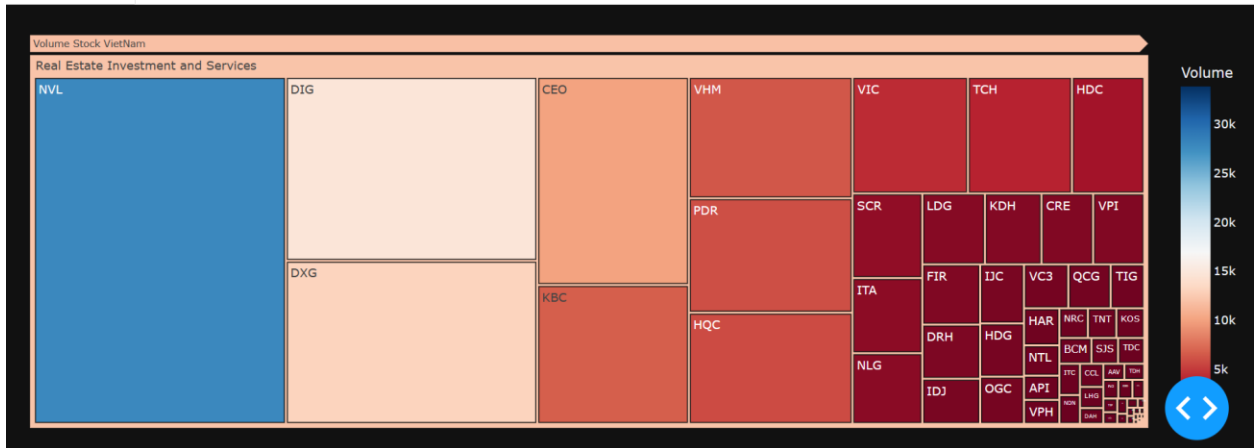
Hình 2.1: Khối lượng giao dịch của các cổ phiếu ở Việt Nam ngày 06/11/2023

Nhận xét: Biểu đồ Treemap thể hiện khối lượng giao dịch của các mã cổ phiếu ở Việt Nam ngày 06/11/2023.

Select Date

11/06/2023

Select



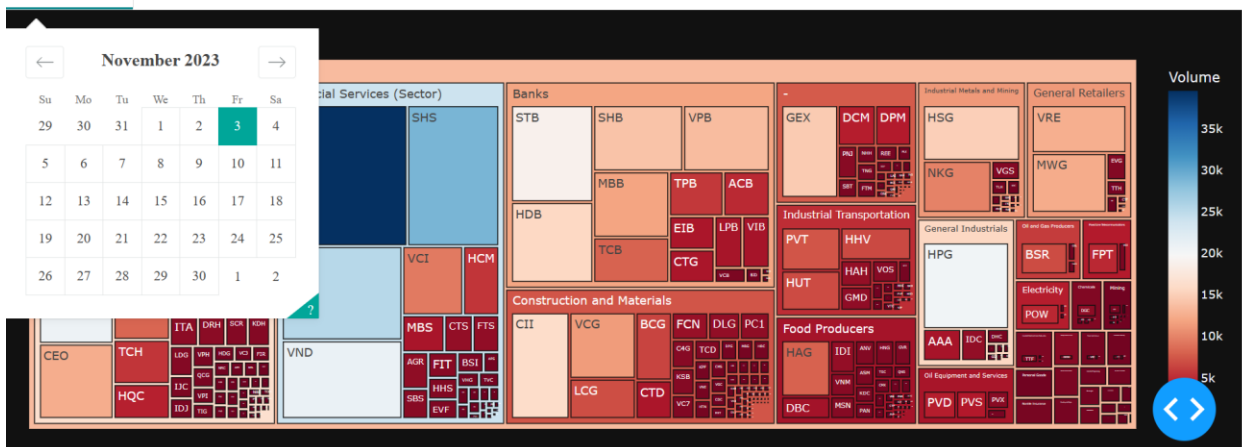
Hình 2.2 : Khối lượng giao dịch của lĩnh vực xây dựng và bất động sản ngày 06/11/2023

Nhận xét Nhìn tổng quan trong ngày 06/11/2023 ở [Hình 2.2] trong lĩnh vực Xây dựng và Bất động sản bạn có thể khối lượng giao dịch của mã cổ phiếu DIG rất là lớn so với các mã cổ phiếu còn lại trong cùng lĩnh vực. Điều này cho ta thấy sự quan tâm và sự tham gia mạnh mẽ của các nhà đầu tư trong việc mua bán cổ phiếu. Khối lượng giao dịch lớn cũng có thể cho thấy mức độ thanh khoản cao của cổ phiếu. Điều này có nghĩa là cổ phiếu DIG có sẵn để mua bán một cách dễ dàng và nhanh chóng, vì có đủ người mua và người bán tham gia giao dịch.

Select Date

11/03/2023|

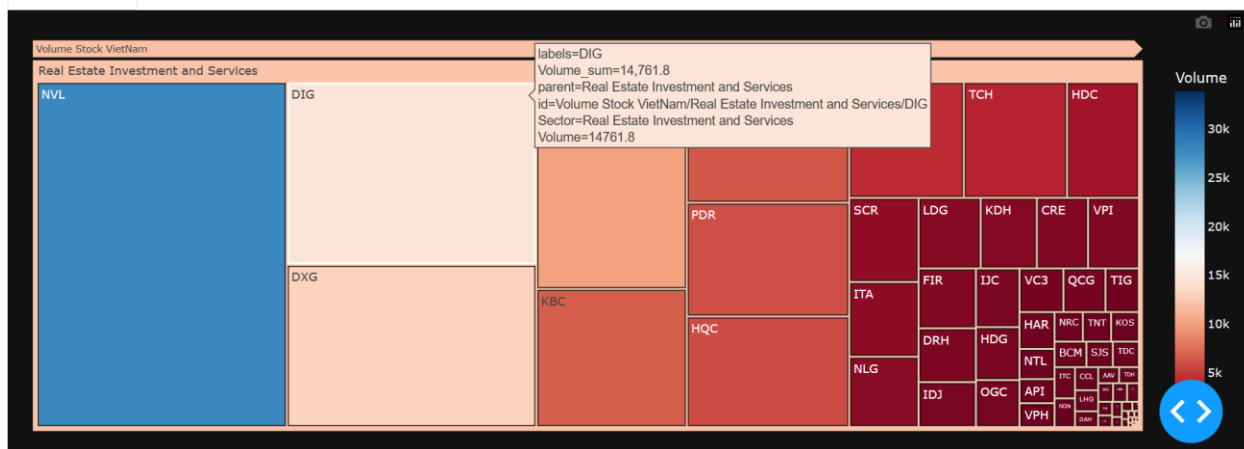
Select



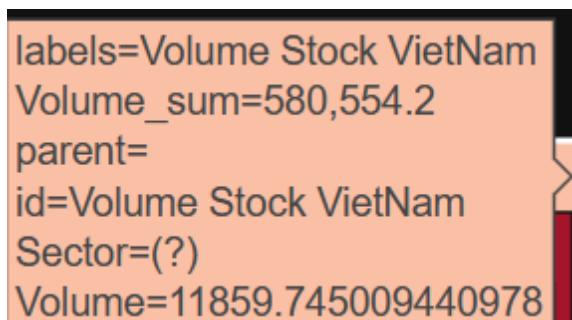
Hình 2.3: Có thể chọn bất kì ngày bạn muốn xem khối lượng giao dịch

Select Date

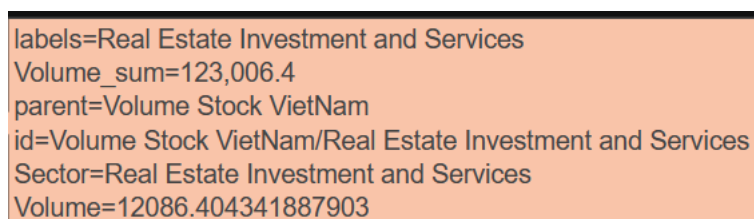
Select



Hình 2.4: Có thể xem khối lượng giao dịch của DIG và lĩnh vực hoạt động



Hình 2.5



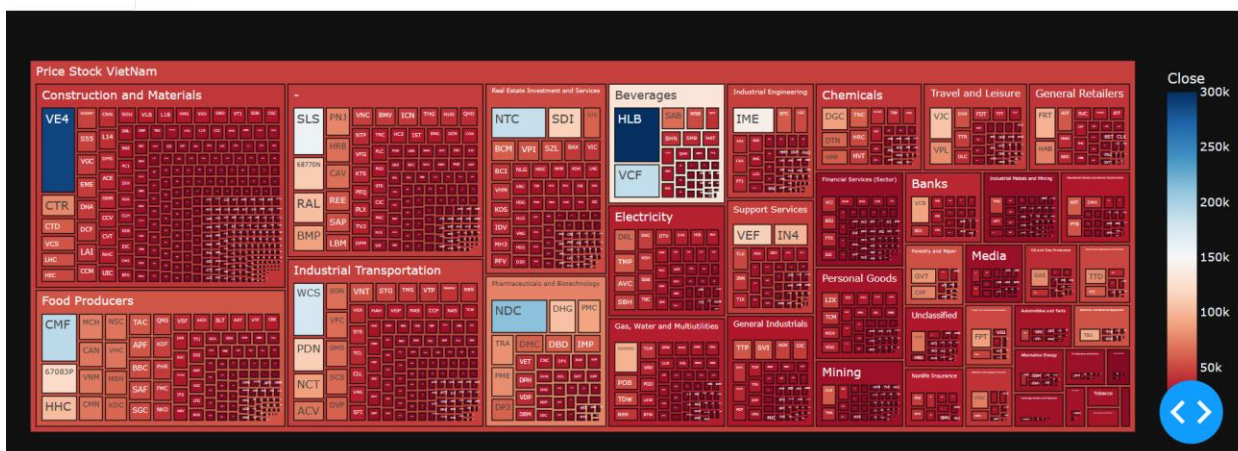
Hình 2.6

Hình 2.5; 2.6: Có thể xem khối lượng giao dịch của Việt Nam và khối lượng giao dịch của từng lĩnh vực trong ngày hôm đó

4. Tổng quan về giá giao dịch của mã cổ phiếu DIG:

Select Date

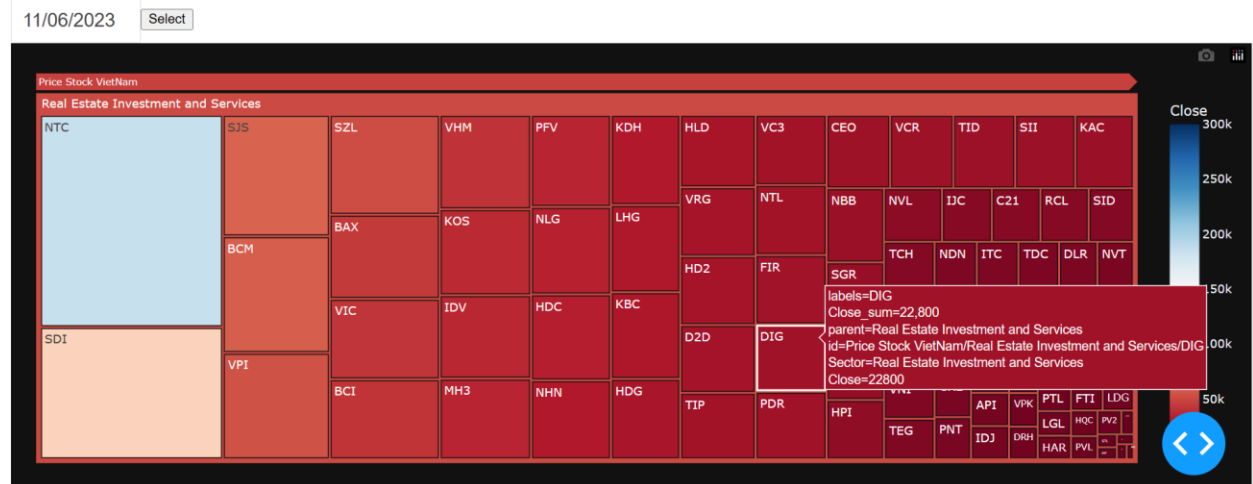
Select



Hình 2.7: Giá giao dịch của các cổ phiếu ở Việt Nam ngày 06/11/2023

Nhận xét: Biểu đồ Treemap thể hiện giá giao dịch của các mã cổ phiếu ở Việt Nam ngày 06/11/2023.

Select Date



Hình 2.8: Giá giao dịch của mã cổ phiếu DIG ở Việt Nam ngày 06/11/2023

Nhận xét: Biểu đồ Treemap thể hiện giá giao dịch của mã cổ phiếu DIG ngày 06/11/2023.

Nhìn tổng quan ngành bất động sản và xây dựng thì giá của mã cổ phiếu DIG đang thuộc top 20 mã cổ phiếu có giá cao nhất của ngành.

5. Phân tích kỹ thuật:

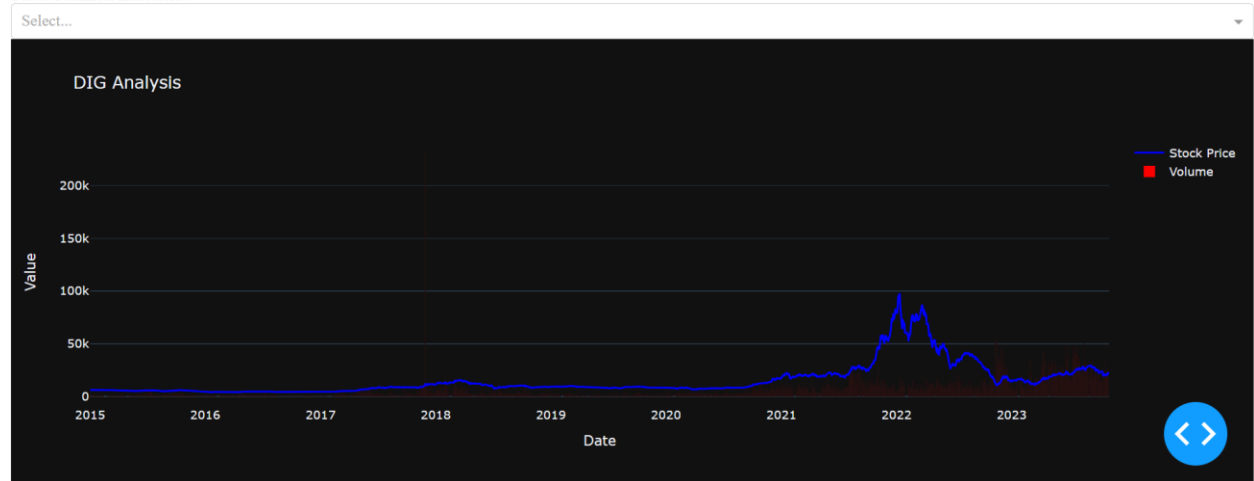
Từ bảng của [Hình 3.1] này sẽ vẽ các đường chỉ báo phân tích kỹ thuật về mã cổ phiếu.

	Date	Close	Volume	
0	2015-01-01	6211.05	NaN	2303 2023-10-31 20200.00 24265.6
1	2015-01-02	6211.05	NaN	2304 2023-11-01 20600.00 19774.8
2	2015-01-05	6163.64	567.4	2305 2023-11-02 22000.00 26227.8
3	2015-01-06	6211.05	1077.7	2306 2023-11-03 22300.00 26869.4
4	2015-01-07	6163.64	959.6	2307 2023-11-06 22800.00 14761.8
...	[2308 rows x 3 columns]

Hình 3.1: Bảng ngày, giá đóng cửa , khối lượng giao dịch của mã cổ phiếu DIG

DIG Analysis Dashboard

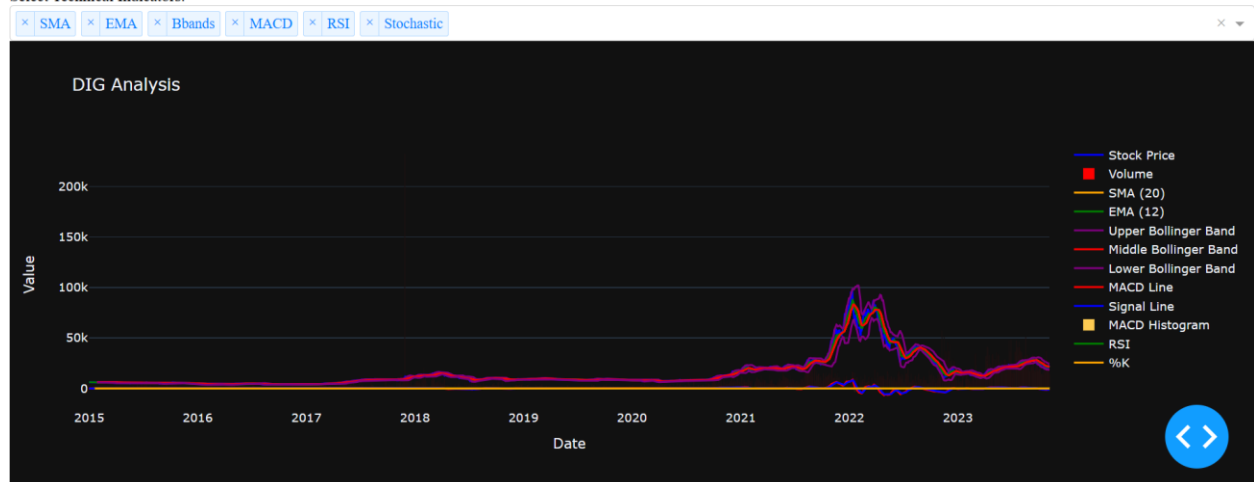
Select Technical Indicators:



Hình 3.2: Biểu đồ giá và khối lượng giao dịch của DIG

DIG Analysis Dashboard

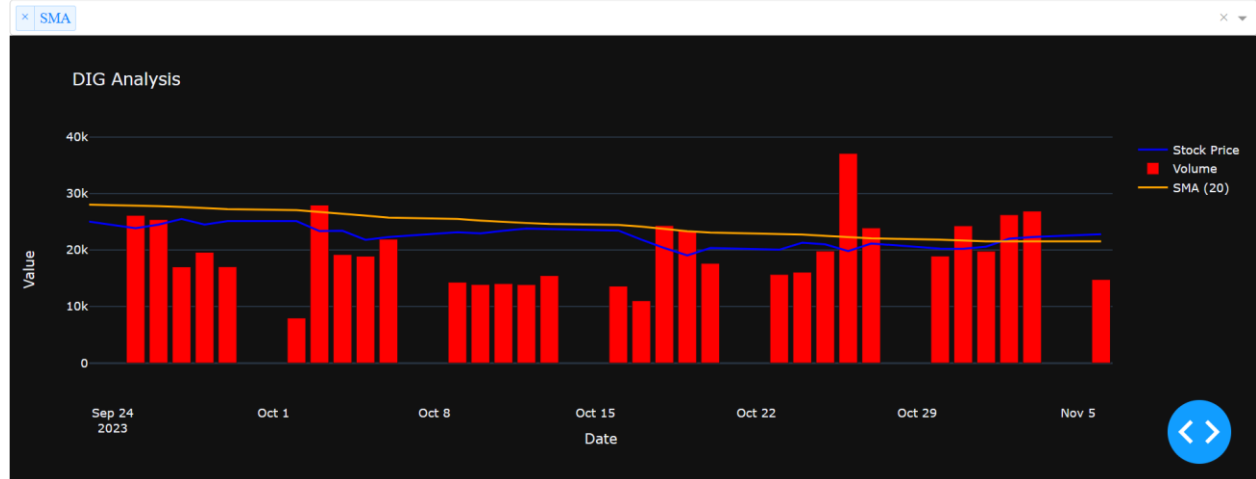
Select Technical Indicators:



Hình 3.3: Các đường chỉ báo SMA, EMA, Bband, MACD, RSI, %K của DIG

DIG Analysis Dashboard

Select Technical Indicators:

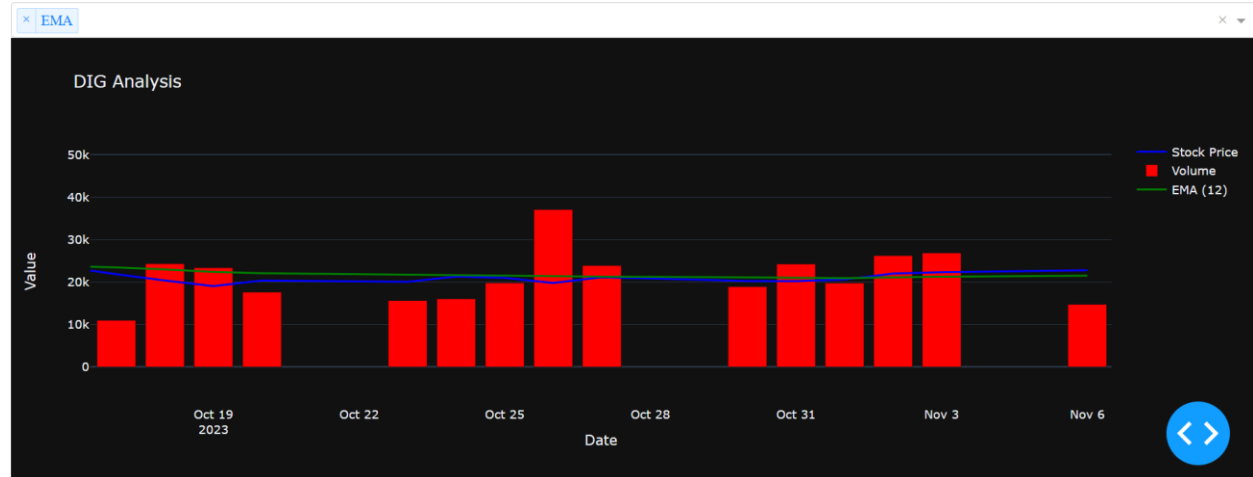


Hình 3.4: Đường chỉ báo SMA của DIG

Nhận xét: Sử dụng SMA(20) để tính toán giá trung bình đơn giản của 20 phiên gần nhất. Từ [Hình 3.4] ta có thể thấy đường SMA (20) trong khoảng thời gian từ ngày 24/9 đến ngày 5/11. Đường SMA(20) tương đối trơn tru so với đường giá cổ phiếu, có nhiều biến động hơn. Đường giá cổ phiếu cắt đường SMA(20) và đi lên, đây có thể là tín hiệu mua (tín hiệu tăng giá).

DIG Analysis Dashboard

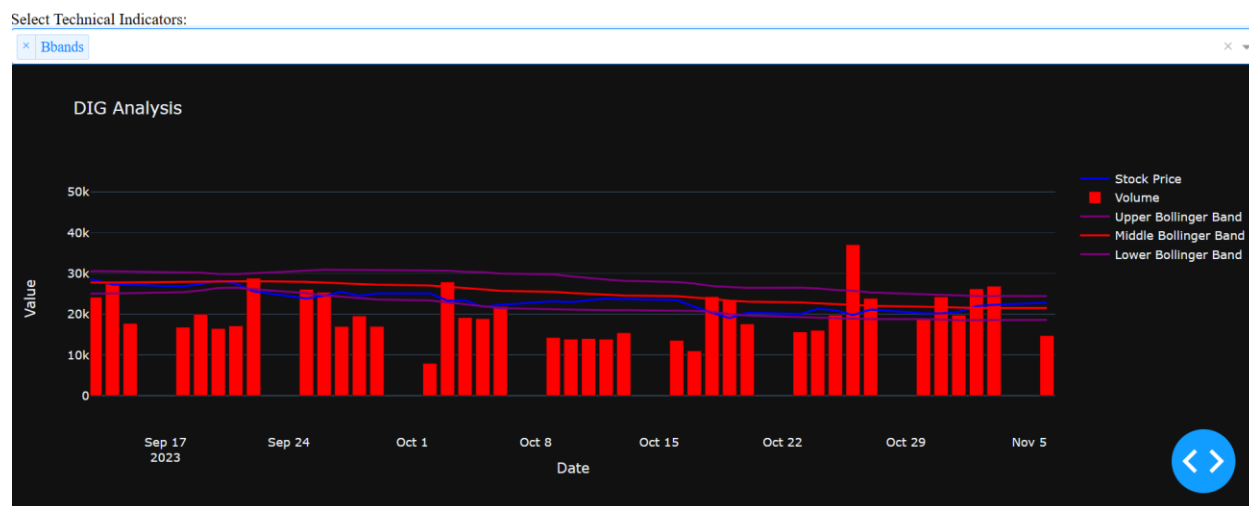
Select Technical Indicators:



Hình 3.5: Đường chỉ báo EMA của DIG

Nhận xét: Sử dụng EMA(12) để tính toán đường trung bình cộng lũy thừa của 12 phiên gần nhất. Từ [Hình 3.5] trong khoảng thời gian từ ngày 9/10/2023 đến 6/11/2023. Giá cổ phiếu dao động trên và dưới đường EMA cho thấy sự biến động về giá. Đường giá cổ phiếu cắt đường EMA(12) và đi lên, đây có thể là tín hiệu mua (tín hiệu tăng giá).

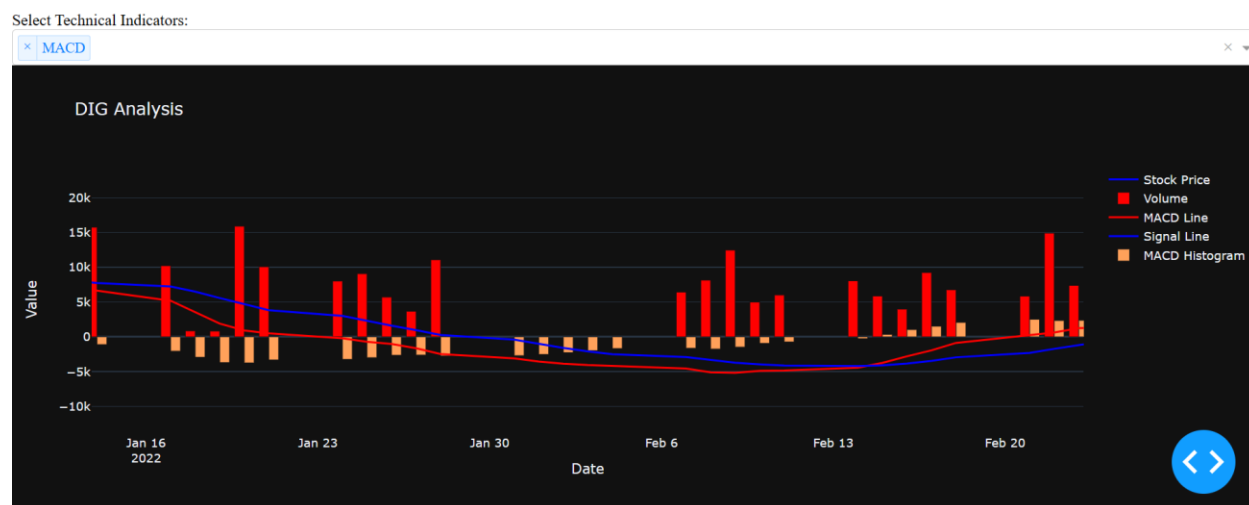
DIG Analysis Dashboard



Hình 3.6: Đường chỉ báo Bband của DIG

Nhận xét : Từ [Hình 3.6] ngày 15/10/2023 đến 22/10/2023 giá chứng khoán đi qua đường dưới của Bollinger Bands và sau đó quay đầu và tăng lên trên đường trung bình di động. Điều này có thể cung cấp tín hiệu mua (Buy entry).

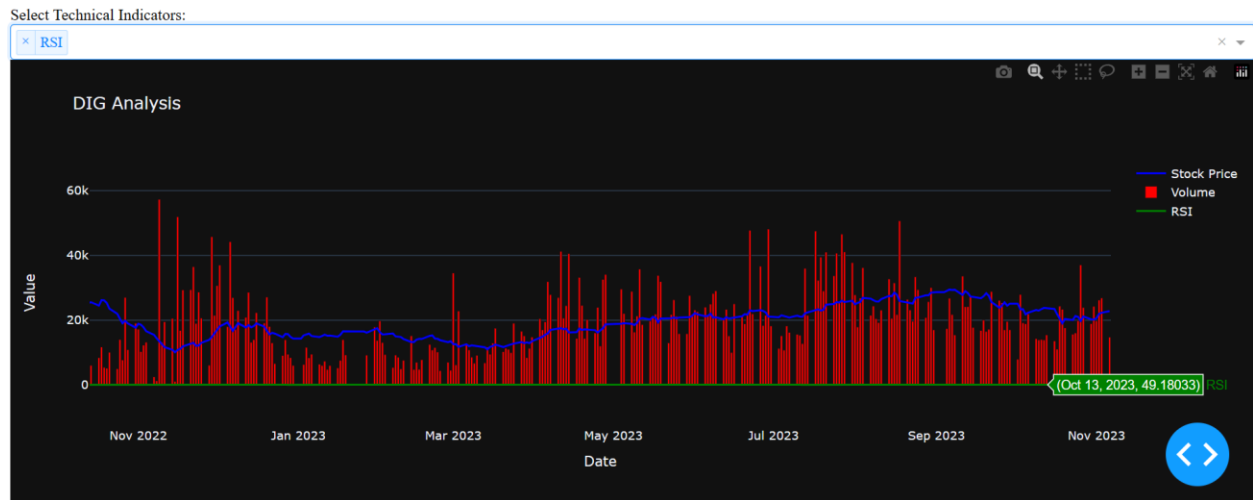
DIG Analysis Dashboard



Hình 3.7: Đường chỉ báo MACD của DIG

Nhận xét : Từ [Hình 3.7] ngày 16/01/2022 đến 20/01/2022 đường MACD Line cắt lên trên đường Signal line từ dưới lên trên, tín hiệu này cho thấy sự thay đổi trong đà tăng giá (tín hiệu mua). Histogram MACD biểu thị sự chênh lệch giữa hai đường này và giúp nhà giao dịch nhìn thấy sự tăng giảm của độ mạnh yếu của xu hướng.

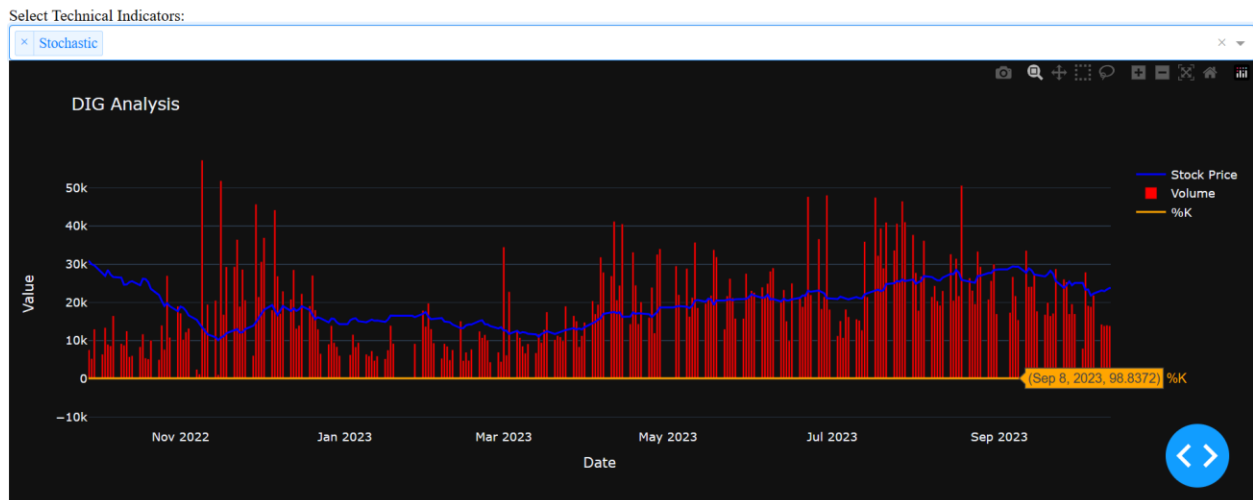
DIG Analysis Dashboard



Hình 3.8: Đường chỉ báo RSI của DIG

Nhận xét : Từ [Hình 3.8] Ngày 13/10/2023 $RSI = 49,18$ cho thấy tài sản đang ở mức trung bình và không ở tình trạng quá mua hay quá bán. Trong phạm vi từ 0 đến 100, một giá trị RSI xấp xỉ 50 thường được coi là tài sản đang ở trạng thái cân bằng.

DIG Analysis Dashboard



Hình 3.9: Đường chỉ báo Stochastic của DIG

Nhận xét : Từ [Hình 3.9] Ngày 8/9/2023 $\%K = 98,8372$ cho thấy tài sản đang ở mức quá mua. Với giá trị Stochastic gần 100, tài sản được coi là đang ở trạng thái quá mua đồng nghĩa với việc giá có thể có xu hướng điều chỉnh hoặc giảm trong tương lai gần.

6. Phân tích báo cáo tài chính CTCP đầu tư phát triển xây dựng:

6.1. Xem các bảng báo cáo tài chính

Bảng DIG

Cân Đối Kế Toán					
DIG	2018	2019	2020	2021	2022
TÀI SẢN NGẮN HẠN	5633632858566	7130818433089	7299152349851	11214616631891	10877019373058
Tiền và tương đương tiền	734834226232	593928466557	402938011038	1000254949868	245914234531
Đầu tư tài chính ngắn hạn	800000000	225000000000	137000000000	2737903082422	176857553417
Các khoản phải thu ngắn hạn	1698828538046	195553803522	2255191037837	3433980916361	4336772938536
Hàng tồn kho, ròng	3132716205480	4193597789629	4395131702337	3844295963880	5923303752238
Tài sản ngắn hạn khác	66453888808	162738373381	108891598639	198181719360	194170894336
TÀI SẢN DÀI HẠN	1198475308854	1066410074669	4527010691862	5632288008979	3870776854827
Phải thu dài hạn	98981612972	84608901281	3111277967009	4110287408009	2381712421430
Tài sản cố định	360299977621	333908838505	764345204814	724974218266	717459864599
GTCL TSCĐ hữu hình	340321014030	314221282162	745785868652	701883536637	674360239308
GTCL Tài sản thuê tài chính	0	0	0	0	0
GTCL tài sản cố định vô hình	19978963591	19687556343	18559336162	23090681629	43099625291
Xây dựng cơ bản dở dang (trước 2015)	0	0	0	0	0
Giá trị ròng tài sản đầu tư	160288894055	155826457411	152042667199	106656556640	10376743998
Tài sản dở dang dài hạn	90943623579	89440203454	87977366707	115145030948	1266523467

Hình 4.1: Bảng cân đối kế toán của DIG của DIG từ năm 2018 đến 2022

Nhận xét : Từ [Hình 4.1] bạn có thể xem được bảng cân đối kế toán của DIG qua các năm và kéo xuống để xem toàn bộ bảng.

Bảng DIG

Kết Quả Kinh Doanh					
DIG	2018	2019	2020	2021	2022
Doanh thu bán hàng và cung cấp dịch vụ	2459138866544	2139478786862	2503768921575	2585736268742	1930222593732
Doanh thu thuần	2345008145519	2115744373631	2487174015553	2568688844439	1896688722277
Lợi nhuận gộp và bán hàng và cung cấp dịch vụ	564583556386	640665017998	640468221409	839490129484	632311466552
Doanh thu hoạt động tài chính	154963084314	197545575808	26858495762	36549319322	86806112261
Chi phí tài chính	-60482178639	-26619168032	-56725752191	-107050252422	-264888595709
Trong đó: Chi phí lãi vay	-21663032812	-37280602441	-37386634438	-96056007775	-243164900279
Lãi/lỗ từ công ty liên doanh	-69176905626	-13888785170	-29014157522	66710007939	5809044858
Chi phí bán hàng	-107969204566	-114152311617	-144047086829	-197134334653	-102059681942
Chi phí quản lý doanh nghiệp	-142520178119	-150401335317	-150149648357	-154539356736	-173817663571
Lợi nhuận thuần từ hoạt động kinh doanh	339398173750	533148993670	287390072272	484025512934	184160682440

Hình 4.2: Bảng kết quả kinh doanh của DIG từ năm 2018 đến 2022

Nhận xét : Từ [Hình 4.2] bạn có thể xem được bảng kết quả kinh doanh của DIG qua các năm và kéo xuống để xem toàn bộ bảng.

Bảng DIG

Lưu Chuyển Tiền Tệ					
DIG	2018	2019	2020	2021	2022
Lãi trước thuế	428283340094	471281200545	899845087070	1281556948813	198787024601
Khấu hao TSCĐ	22806617100	37440918820	44886569674	51221152288	64798586744
Lãi/(lỗ) trước những thay đổi vốn lưu động	336202328095	360351817586	377031434359	533326718092	438816929164
Lưu chuyển tiền tệ ròng từ các hoạt động sản xuất kinh doanh (TT)	430577636100	-245407250565	-504296440297	-1966481277659	-2205947002535
Tiền chi để mua sắm, xây dựng TSCĐ và các tài sản dài hạn khác (TT)	-55484135322	-49904599042	-11580836846	-25989397217	-39738082123
Tiền thu từ thanh lý, nhượng bán TSCĐ và các tài sản dài hạn khác (TT)	6124050907	9400950347	3313154432	12592729982	2330252400
Tiền chi cho vay, mua các công cụ nợ của đơn vị khác (TT)	-232900000000	-278900000000	-127000000000	-3212059682175	-56911200000

Hình 4.3: Bảng lưu chuyển tiền tệ của DIG từ năm 2018 đến 2022

Nhận xét : Từ [Hình 4.3] bạn có thể xem được bảng lưu chuyển tiền tệ của DIG qua các năm và kéo xuống để xem toàn bộ bảng.

Bảng DIG

Thuyết Minh					
DIG	2018	2019	2020	2021	2022
Tiền	734834226232	818928466557	402938011038	1000254949868	245914234531
Tiền mặt	3496558586	4444189710	6519748484	1225970907	1616193984
Tiền gửi Ngân hàng	396117449246	470936576362	282486339253	379556440702	234807825128
Tiền đang chuyển	96020852	43180000	0	0	0
Tiền và tương đương tiền	335124197548	343504520485	113931923301	619472538259	9490215419
Đầu tư tài chính NH	800000000	225000000000	1370000000000	2737903082422	176857553417
Chứng khoán đầu tư ngắn hạn	0	0	0	0	0
Đầu tư nắm giữ đến ngày đáo hạn	800000000	225000000000	1370000000000	2737903082422	176857553417
Đầu tư ngắn hạn	800000000	225000000000	1370000000000	2737903082422	176857553417
Đầu tư dài hạn	0	0	0	0	0
Đầu tư NH khác	0	0	0	0	0
Dự phòng giảm giá ĐTNH	0	0	0	0	0
Hàng tồn kho	3147381338237	4203273070739	4401218373486	3848208105002	59257372578

Hình 4.4: Bảng thuyết minh của DIG từ năm 2018 đến 2022

Nhận xét : Từ [Hình 4.4] bạn có thể xem được bảng thuyết minh của DIG qua các năm và kéo xuống để xem toàn bộ bảng.

Bảng DIG

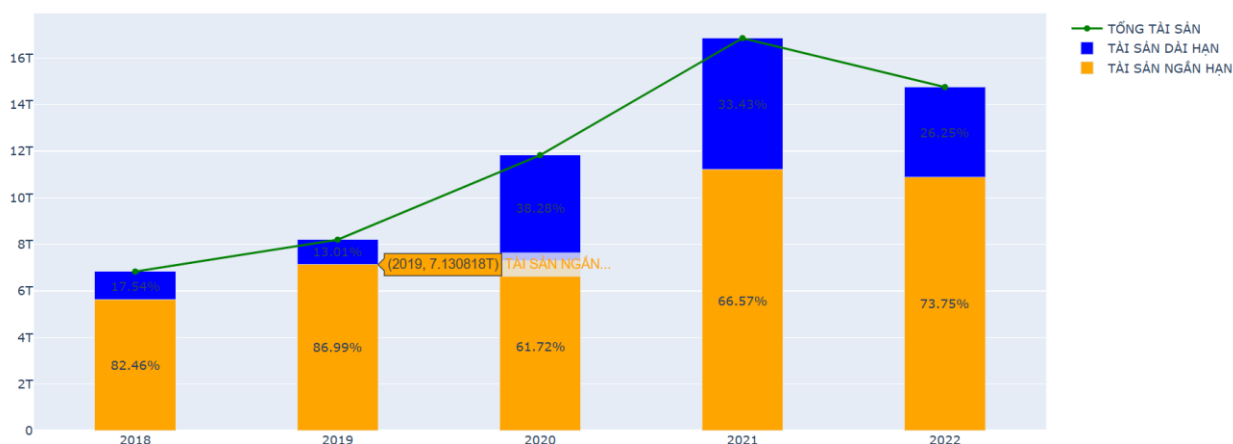
DIG	2018	2019	2020	2021	2022
BCTCKH. Doanh thu kế hoạch	2900000000000	2500000000000	2800000000000	5000000000000	4000000000000
BCTCKH. Tổng lợi nhuận kế toán trước thuế	5500000000000	6500000000000	14440000000000	19000000000000	14000000000000
BCTCKH. Lợi nhuận sau thuế thu nhập doanh nghiệp	4400000000000	5200000000000	11552000000000	15200000000000	11200000000000

Hình 4.5: Bảng kết quả kinh doanh của DIG từ năm 2018 đến 2022

Nhận xét : Từ [Hình 4.5] bạn có thể xem được bảng kết quả kinh doanh của DIG qua các năm.

6.2. Phân tích bảng cân đối kế toán

Biểu đồ Tài Sản của DIG theo các năm



Hình 5.1: Biểu đồ biến động kết cấu tài sản của DIG từ năm 2018 đến 2022

Nhận xét : Từ [Hình 5.1] có thể xem được tài sản ngắn hạn, tài sản dài hạn theo phần trăm tài sản của DIG qua các năm. Bạn cũng có thể xem được giá trị của tài sản ngắn hạn và tài sản dài hạn bằng cách chỉ chuột vào cột bạn muốn xem, ví dụ ở đây bạn có thể thấy tài sản ngắn hạn của năm 2019 là 7130818 (đơn vị: triệu).

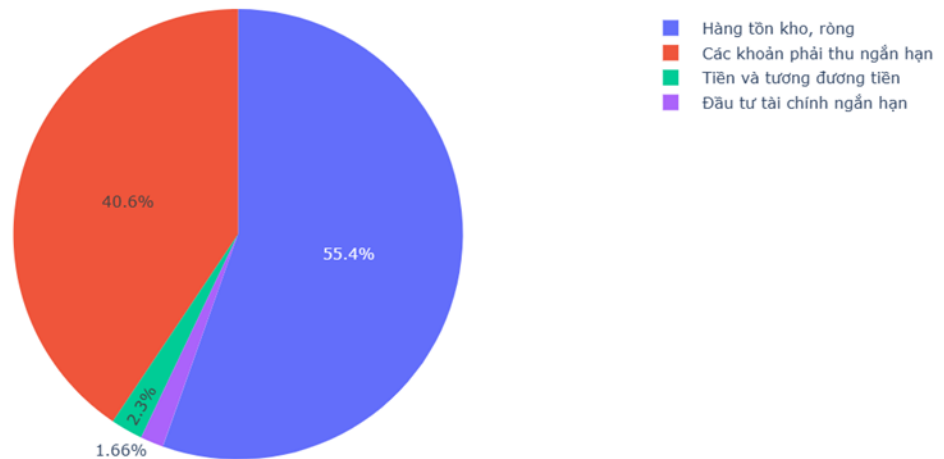
Từ [Hình 5.1] bạn có thể thấy tổng tài sản năm 2022 đạt gần 15 nghìn tỷ đồng, tỷ trọng chính đến từ tài sản ngắn hạn, chiếm tỷ trọng hơn 73,75% và tài sản dài hạn chiếm 26,35%.

Tài sản của DIG tăng lên qua các năm, từ khoảng 7.000 tỷ năm 2018 lên gần 15.000 tỷ năm 2022. Tình hình tài sản năm 2022 của doanh nghiệp đang giảm khoảng 2000 tỷ so với năm 2021.

Tài sản ngắn hạn chiếm phần lớn tài sản của DIG, dao động từ 61,72% đến 86,99% trong 5 năm. Tài sản dài hạn có sự biến động lớn hơn, từ 17,54% năm 2018 lên 38,28% năm 2020, rồi giảm xuống 26,25% năm 2022. Biểu đồ này có thể cho thấy

DIG đang đầu tư vào các loại tài sản khác nhau để tăng hiệu quả hoạt động và tối ưu hóa nguồn vốn.

TÀI SẢN NGẮN HẠN 2022



Hình 5.2: Biểu đồ tài sản ngắn hạn của DIG năm 2022

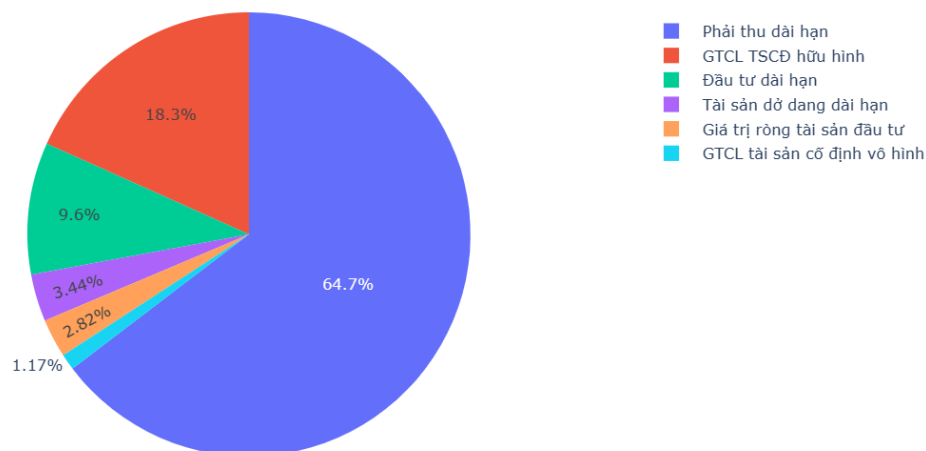
Nhận xét : Từ [Hình 5.2] biểu đồ tài sản ngắn hạn của DIG năm 2022 có thể thấy:

Hàng tồn kho rỗng chiếm 55.4%: Tỷ lệ này cho thấy DIG có một phần lớn tài sản ngắn hạn được đầu tư vào hàng tồn kho, có thể là các vật liệu xây dựng hay các sản phẩm đã hoàn thiện sẵn sàng để bán ra thị trường. Điều này cho thấy công ty có hoạt động kinh doanh tích cực và có kế hoạch phát triển trong ngành xây dựng và bất động sản.

Các khoản phải thu ngắn hạn chiếm 40.6%: Tỷ lệ này đại diện cho các khoản tiền mà DIG đang chờ thu từ khách hàng, đối tác hoặc các bên liên quan trong thời gian ngắn hạn. Điều này có thể bao gồm tiền bán hàng, công nợ khách hàng và các khoản thanh toán chưa được nhận. Việc có một tỷ lệ cao cho thấy công ty đang có một lượng lớn tiền mà họ có thể thu được trong tương lai gần.

Tiền và tương đương tiền chiếm 2.3%: Tài sản tiền và tương đương tiền thường là các khoản tiền mặt hoặc tương đương có thể sử dụng ngay lập tức. Tỷ lệ này cho thấy DIG có một phần nhỏ tài sản dễ dàng chuyển đổi thành tiền mặt trong ngắn hạn.

Đầu tư tài chính ngắn hạn chiếm 1.66%: Đây có thể là các khoản đầu tư ngắn hạn của DIG vào các công ty khác hoặc các công cụ tài chính có thể chuyển đổi thành tiền mặt trong thời gian ngắn. Tuy tỷ lệ này thấp, nhưng nó có thể là một phần của chiến lược tài chính của công ty hoặc một phần nhỏ của tổng tài sản ngắn hạn.



Hình 5.3: Biểu đồ tài sản dài hạn của DIG năm 2022

Nhận xét : Từ [Hình 5.3] biểu đồ tài sản dài hạn của DIG năm 2022 có thể thấy:

Phải thu dài hạn chiếm tỷ trọng lớn (64.7%): Tỷ lệ này cho thấy DIG có một số lượng đáng kể các khoản phải thu từ khách hàng hoặc các bên liên quan mà công ty dự kiến thu hồi trong thời gian dài hơn một năm. Điều này có thể ám chỉ rằng DIG có các dự án hoặc hợp đồng dài hạn và đòi hỏi thời gian để thu hồi tiền. Quản lý kỹ càng và hiệu quả của các khoản phải thu dài hạn là quan trọng để đảm bảo rằng công ty có đủ tiền mặt để hoạt động và đáp ứng các cam kết tài chính.

Tài sản cố định hữu hình chiếm tỷ trọng đáng kể (18.3%): Tỷ lệ này chỉ ra rằng DIG có một lượng lớn tài sản cố định hữu hình như đất, nhà xưởng, thiết bị, và công trình xây dựng. Đây là các tài sản dài hạn mà công ty sử dụng để thực hiện hoạt động kinh doanh. Việc có một tỷ trọng cao trong tài sản cố định hữu hình có thể là dấu hiệu của sự đầu tư và phát triển trong lĩnh vực xây dựng hoặc bất động sản.

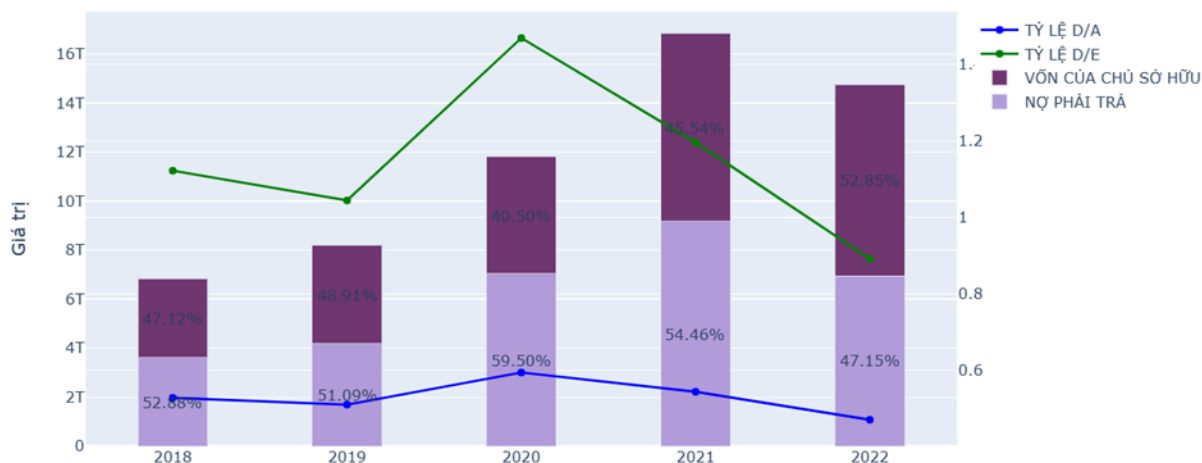
Đầu tư dài hạn chiếm tỷ trọng nhỏ (9.6%): Tỷ lệ này cho thấy DIG có một số lượng đầu tư dài hạn như cổ phiếu, trái phiếu, hoặc các khoản đầu tư khác. Tuy nhiên, tỷ trọng này không lớn so với các thành phần khác trong tài sản dài hạn của công ty.

Tài sản dở dang dài hạn chiếm tỷ trọng thấp (3.44%): Tài sản dở dang dài hạn là những dự án hoặc công việc mà công ty đã bắt đầu nhưng chưa hoàn thành và chưa thu hồi được tiền. Tỷ lệ này cho thấy tỷ trọng tài sản dở dang dài hạn của DIG không cao, điều này có thể có ý nghĩa tích cực về việc quản lý dự án và tiến độ hoàn thành của công ty.

Giá trị ròng tài sản đầu tư chiếm tỷ trọng thấp (2.82%): Tỷ lệ này cho thấy giá trị ròng của các khoản đầu tư của DIG sau khi trừ đi các nợ và nghĩa vụ tài chính không quá lớn so với tổng tài sản dài hạn. Điều này có thể yêu cầu công ty tập trung vào việc quản lý và tối ưu hóa lợi nhuận từ các khoản đầu tư của mình.

Tài sản cố định vô hình chiếm tỷ trọng thấp (1.17%): Tỷ lệ này khá thấp so với các thành phần khác trong tài sản dài hạn của công ty. Tài sản cố định vô hình thường liên quan đến các quyền sở hữu trí tuệ như thương hiệu, bản quyền, giấy phép, và các tài sản không vật thể khác. Một tỷ lệ thấp có thể cho thấy công ty không có nhiều đầu tư vào các quyền sở hữu trí tuệ hoặc không có một số lớn các tài sản vô hình quan trọng trong hoạt động kinh doanh của mình.

BIỂU ĐỒ BIẾN ĐỘNG NGUỒN VỐN



Hình 5.4: Biểu đồ biến động nguồn vốn của DIG từ năm 2018 đến 2022

Nhận xét : Từ hình [5.4] có thể thấy:

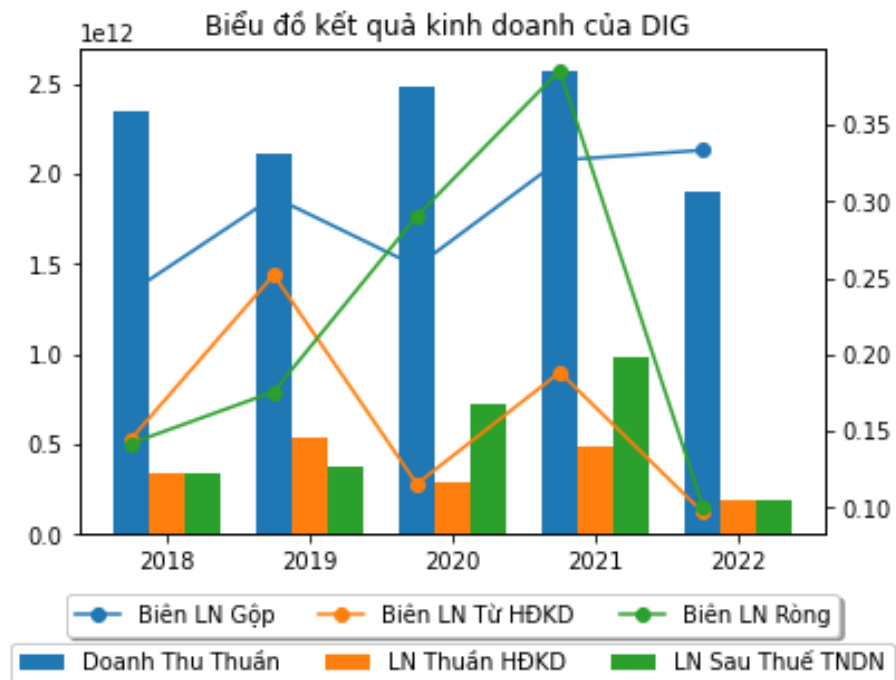
Biến động lớn trong vốn chủ sở hữu và nợ phải trả cho thấy có sự biến động đáng kể trong cả vốn chủ sở hữu và nợ phải trả qua các năm. Điều này có thể chỉ ra rằng công ty đã trải qua các thay đổi quan trọng trong cấu trúc tài chính của mình, bao gồm việc tăng vốn, giảm nợ hoặc thay đổi trong việc sử dụng các nguồn tài chính.

Năm 2021 có sự tăng vọt về vốn chủ sở hữu và giảm tỷ lệ nợ trên vốn chủ sở hữu, điều này có thể cho thấy công ty đã tăng vốn chủ sở hữu hoặc giảm nợ để cải thiện cấu trúc tài chính và năng lực thanh toán.

Tỷ lệ D/A và D/E giảm dần từ năm 2020 đến 2022, điều này có thể cho thấy công ty đã giảm sự phụ thuộc vào nợ và tăng cường vốn chủ sở hữu để cải thiện cấu trúc tài chính và giảm rủi ro.

Năm 2022 cho thấy sự giảm nhẹ về vốn chủ sở hữu so với năm trước đó. Điều này có thể chỉ ra rằng công ty đã có sự thay đổi trong cấu trúc tài chính hoặc việc sử dụng nguồn vốn khác.

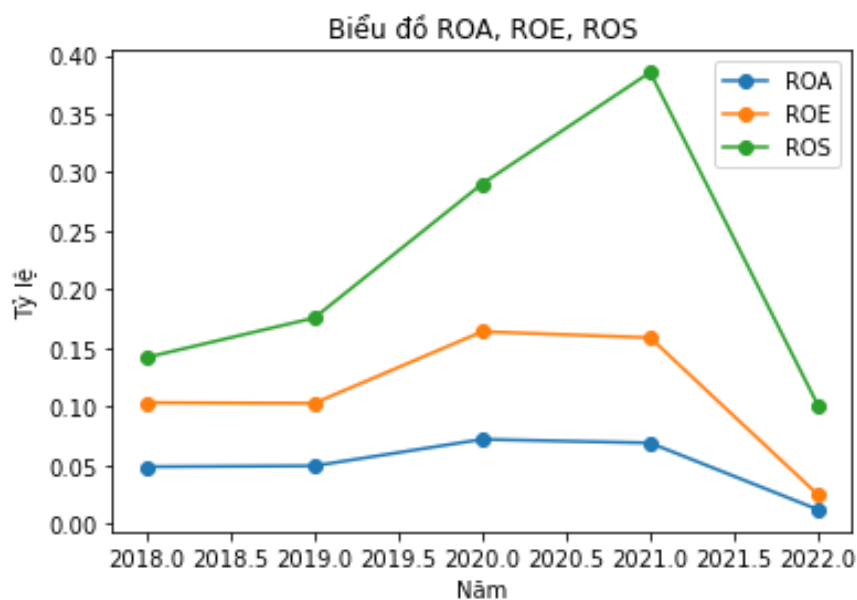
6.3. Phân tích kết quả hoạt động kinh doanh



Hình 6.1: Biểu đồ kết quả kinh doanh của DIG từ năm 2018 đến 2022

Nhận xét : Biểu đồ kết quả kinh doanh của DIG cho thấy doanh nghiệp này có sự tăng trưởng ổn định trong lĩnh vực bất động sản và xây dựng trong giai đoạn 2018-2021. Năm 2022, có sự sụt giảm về tăng trưởng so với các năm trước.

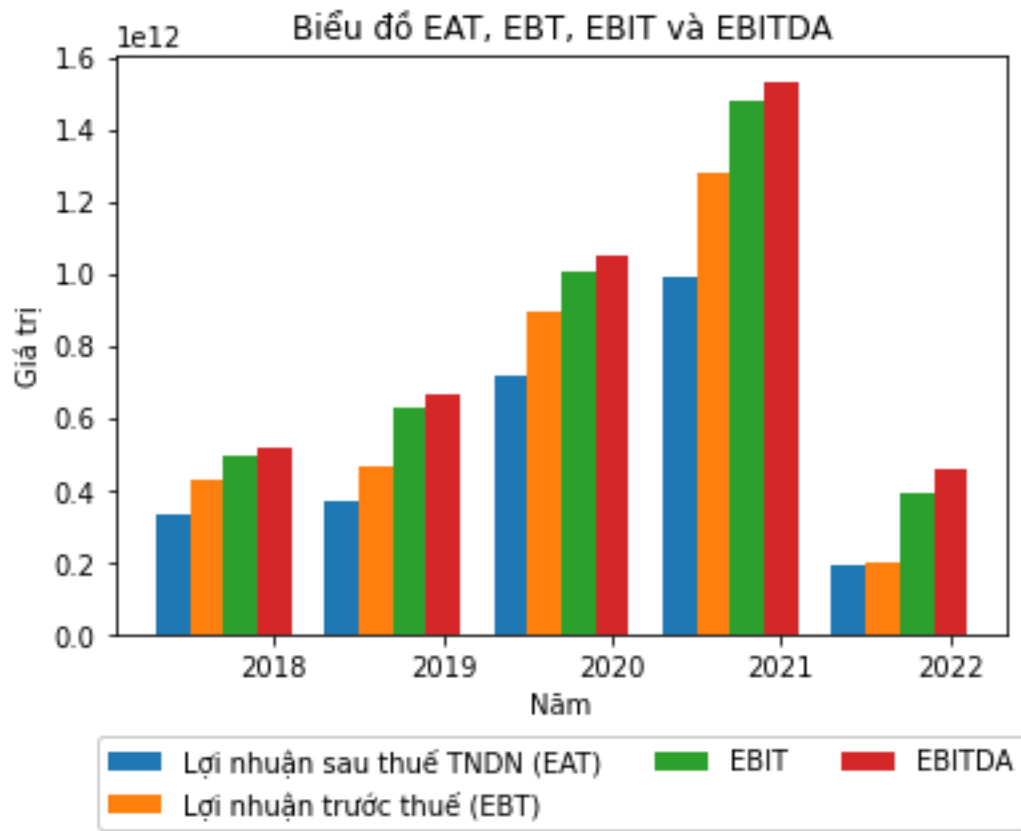
Có thể thấy, DIG là một doanh nghiệp bất động sản có tiềm năng phát triển tốt. Doanh nghiệp này có lợi thế về quỹ đất lớn, kinh nghiệm triển khai dự án và năng lực tài chính vững mạnh.



Hình 6.2: Biểu đồ đánh giá hiệu suất kinh doanh của DIG từ năm 2018 đến 2022

Nhận xét : Trong giai đoạn 2018-2020, lợi nhuận trên tổng tài sản, vốn chủ sở hữu và doanh thu thuần của DIG tăng trưởng với tốc độ khá ổn định. Tuy nhiên, trong năm 2021, lợi nhuận đã tăng đột biến. Nguyên nhân của mức tăng trưởng này có thể là do nhu cầu về bất động sản tăng cao trong bối cảnh dịch bệnh COVID-19, bên cạnh đó, DIG cũng có kế hoạch đầu tư vào các dự án bất động sản lớn, tiềm năng.

Trong giai đoạn 2022, tăng trưởng DIG có xu hướng chững lại, với mức tăng trưởng không cao như trước. Nguyên nhân có thể là do thị trường chứng khoán Việt Nam đang trong giai đoạn điều chỉnh, bên cạnh đó, một số yếu tố rủi ro khác như lạm phát, lãi suất tăng,... cũng có thể tác động đến giá cổ phiếu của DIG.

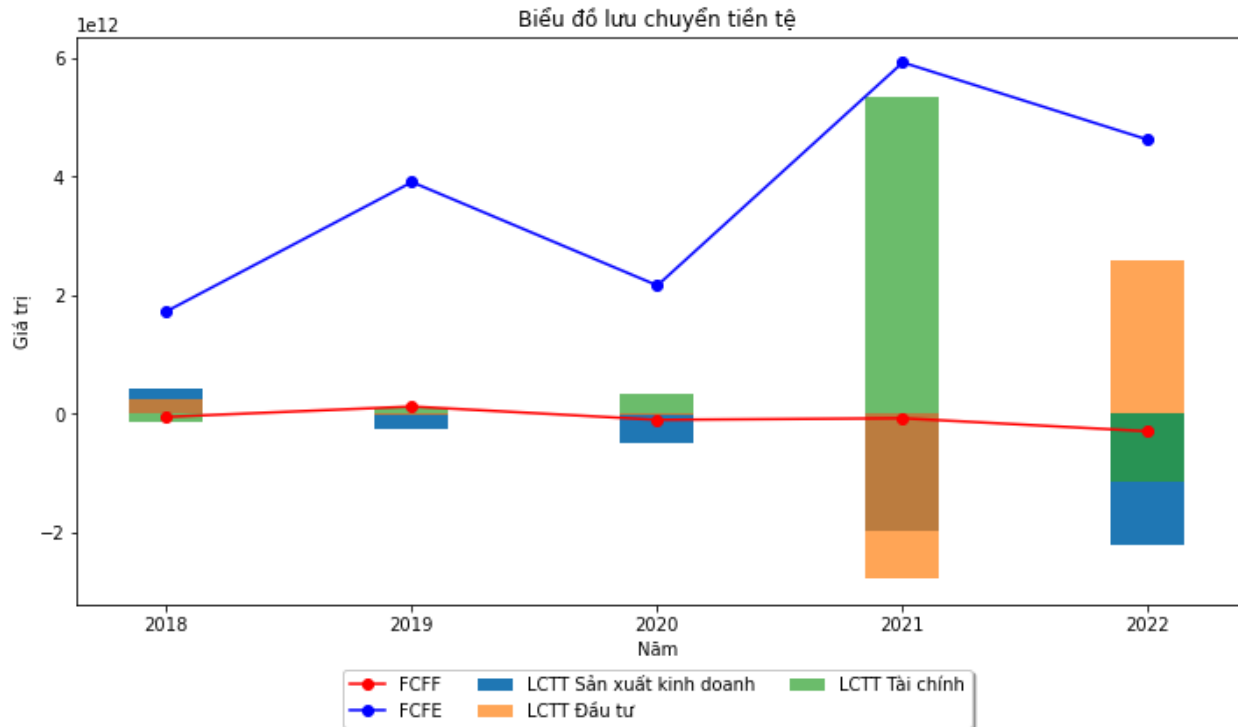


Hình 6.3: Biểu đồ lợi nhuận của DIG từ năm 2018 đến 2022

Nhận xét : Biểu đồ thể hiện tình hình lợi nhuận của Công ty Cổ phần Đầu tư Phát triển Xây dựng (DIG) trong giai đoạn 2018-2022.

Nhìn chung, tình hình kinh doanh của DIG trong giai đoạn 2018-2022 có xu hướng tăng trưởng. Tuy nhiên, năm 2022 lợi nhuận của DIG giảm mạnh do ảnh hưởng của dịch bệnh COVID-19 và các yếu tố vĩ mô khác, thị trường bất động sản Việt Nam trong năm 2022 đã có dấu hiệu chững lại. Điều này dẫn đến việc nhu cầu về bất động sản giảm, khiến cho doanh thu và lợi nhuận của các doanh nghiệp bất động sản, trong đó có DIG, bị ảnh hưởng.

6.4. Phân tích bảng lưu chuyển tiền tệ



Hình 7: Biểu đồ lưu chuyển tiền tệ của DIG từ năm 2018 đến 2022

Nhận xét :

Lưu chuyển tiền tệ từ hoạt động kinh doanh (LCTTKD) của DIG có xu hướng giảm dần trong giai đoạn 2018-2022. Sự giảm sút này có thể do một số nguyên nhân, bao gồm: Doanh thu thuần của DIG giảm trong giai đoạn này. Thuế và các khoản phải nộp Nhà nước tăng. Các khoản trả trước cho người bán tăng.

Lưu chuyển tiền tệ từ hoạt động đầu tư (LCTTĐT) của DIG có xu hướng giảm dần trong giai đoạn 2018-2021. Tuy nhiên, trong năm 2022, LCTTĐT đã tăng mạnh. Sự tăng trưởng này là do DIG đã thực hiện các khoản chi lớn cho đầu tư tài sản cố định, như chi phí xây dựng, chi phí mua sắm máy móc thiết bị,...

Lưu chuyển tiền tệ từ hoạt động tài chính (LCTTTC) của DIG có xu hướng tăng trưởng trong giai đoạn 2018-2021. Tuy nhiên năm 2022 có xu hướng giảm nguyên nhân là DIG đã thực hiện các khoản chi lớn cho phát hành trái phiếu và vay vốn ngân hàng trong năm 2022. Các khoản chi này đã làm giảm dòng tiền từ hoạt động tài chính điều này đã làm tăng chi phí tài chính của DIG, dẫn đến giảm lợi nhuận.

6.5. Các chỉ số tài chính

6.5.1. Định giá

	Năm	2018	2019	2020	2021	2022
1	EPS cơ bản	1233.0	1286.0	2157.0	1689.0	236.0
2	P/E	8.844385	6.899152	4.382311	17.07832	185.596676
3	P/B	0.92034	0.642961	0.663005	2.210469	4.581859
4	P/S	1.255654	1.212462	1.271955	6.581776	18.72574

Hình 8.1: Bảng định giá của DIG từ năm 2018 đến 2022

Nhận xét : Từ [Hình 8.1] có thể thấy:

EPS cơ bản của doanh nghiệp có xu hướng tăng trưởng mạnh mẽ trong giai đoạn 2018-2020, từ 1.233 đồng/cổ phiếu năm 2018 lên 2.157 đồng/cổ phiếu năm 2020 và bắt đầu giảm từ năm 2020 đến 2022.

P/E của doanh nghiệp có xu hướng tăng mạnh trong giai đoạn 2018-2022, từ 8,84 năm 2018 xuống 185,6 năm 2022.

P/B của doanh nghiệp có xu hướng tăng mạnh trong giai đoạn 2018-2022, từ 0,92 năm 2018 lên 4,58 năm 2022.

P/S của doanh nghiệp có xu hướng tăng mạnh trong giai đoạn 2018-2022, từ 1,26 lần năm 2018 lên 18,73 năm 2022.

6.5.2. Khả năng sinh lợi

	Năm	2018	2019	2020	2021	2022
1	Tỷ lệ lãi ròng (%)	0.141972	0.175741	0.290248	0.385388	0.100895
2	Tỷ lệ lãi gộp (%)	0.24076	0.302808	0.257508	0.326817	0.333377
3	Tỷ lệ EBIT (%)	0.210975	0.299667	0.405575	0.576068	0.207181
4	Tỷ lệ lãi từ HĐKD (%)	0.144732	0.251991	0.115549	0.188433	0.097096

Hình 8.2: Bảng đánh giá khả năng sinh lợi của DIG từ năm 2018 đến 2022

Nhận xét :

Nhìn chung, tỷ lệ lãi gộp của doanh nghiệp này đang có xu hướng tăng lên. Năm 2018, tỷ lệ lãi gộp là 24,076%, tăng lên 33,3377% vào năm 2022.

Từ năm 2018 – 2021, tỷ lệ lãi ròng của doanh nghiệp này là 14,1972%, tăng lên 38,5388% vào năm 2021. Doanh nghiệp đã cải thiện hiệu quả hoạt động sản xuất, bán hàng và quản lý chi phí. Ngoài ra, tỷ lệ EBIT và tỷ lệ lãi từ HĐKD cũng có xu hướng

tăng lên. Tỷ lệ EBIT là chỉ số phản ánh hiệu quả hoạt động trước lãi vay, thuế và khấu hao. Tỷ lệ lãi từ HĐKD là chỉ số phản ánh hiệu quả hoạt động kinh doanh trước lãi vay và thuế. Sự gia tăng của hai chỉ số này cũng cho thấy doanh nghiệp đang hoạt động hiệu quả hơn.

Tuy nhiên, năm 2022 là một năm khó khăn đối với doanh nghiệp. Tỷ lệ lãi ròng của doanh nghiệp giảm mạnh từ 38,5388% trong năm 2021 xuống còn 10,0895% trong năm 2022.

6.5.3. Sức mạnh tài chính

	Năm	2018	2019	2020	2021	2022
1	Thanh toán hiện hành	1.879534	2.181042	1.219236	2.401195	2.756646
2	Thanh toán nhanh	0.812203	0.848607	0.466893	1.535648	1.206248
3	Thanh toán lãi vay	7.44482	3.895944	9.263678	7.466575	2.023774
4	D/E	1.122195	1.044479	1.468861	1.196043	0.89202
5	D/A	0.52879	0.510878	0.594955	0.544635	0.471464

Hình 8.3: Bảng đánh giá sức mạnh tài chính của DIG từ năm 2018 đến 2022

Nhận xét :

Thanh toán hiện hành (Current Ratio): Trong giai đoạn 2018-2022, chỉ số thanh toán hiện hành của DIG luôn ở mức trên 1, cho thấy công ty có khả năng thanh toán các khoản nợ ngắn hạn tốt.

Thanh toán nhanh (Quick Ratio): Trong giai đoạn 2018-2022, chỉ số thanh toán nhanh của DIG luôn ở mức trên 0,8, cho thấy công ty có khả năng thanh toán các khoản nợ ngắn hạn tốt.

Thanh toán lãi vay (Interest Coverage Ratio): Trong giai đoạn 2018-2022, chỉ số thanh toán lãi vay của DIG có xu hướng giảm, từ 7,44 lần năm 2018 xuống còn 2,02 lần năm 2022. Điều này cho thấy khả năng trả lãi vay của DIG đang giảm.

6.5.4. Hiệu quả quản lý

	Năm	2018	2019	2020	2021	2022
1	ROA	0.048729	0.04948	0.072105	0.06905	0.012114
2	ROE	0.103413	0.102872	0.164075	0.158879	0.024746
3	ROIC	0.081143	0.079229	0.122317	0.085503	0.016795
4	ROCE	0.129014	0.128662	0.172744	0.121524	0.036378

Hình 8.4: Bảng hiệu đánh giá quả quản lý của DIG từ năm 2018 đến 2022

Nhân xét :

Nhìn chung, DIG đang có sự phát triển tích cực về doanh thu, lợi nhuận và các chỉ số sinh lời. Tuy nhiên, chỉ số sinh lời của công ty đã giảm trong năm 2022.

6.5.5. Khả năng hoạt động

	Năm	2018	2019	2020	2021	2022
1	Vòng quay tổng TS	0.343233	0.281549	0.248427	0.179171	0.120064
2	Vòng quay HTK	0.928776	0.752468	0.728313	0.827285	0.517835
3	Vòng quay TSDH	1.95666	1.868301	0.889321	0.505682	0.399174

Hình 8.5: Bảng đánh giá khả năng hoạt động của DIG từ năm 2018 đến 2022

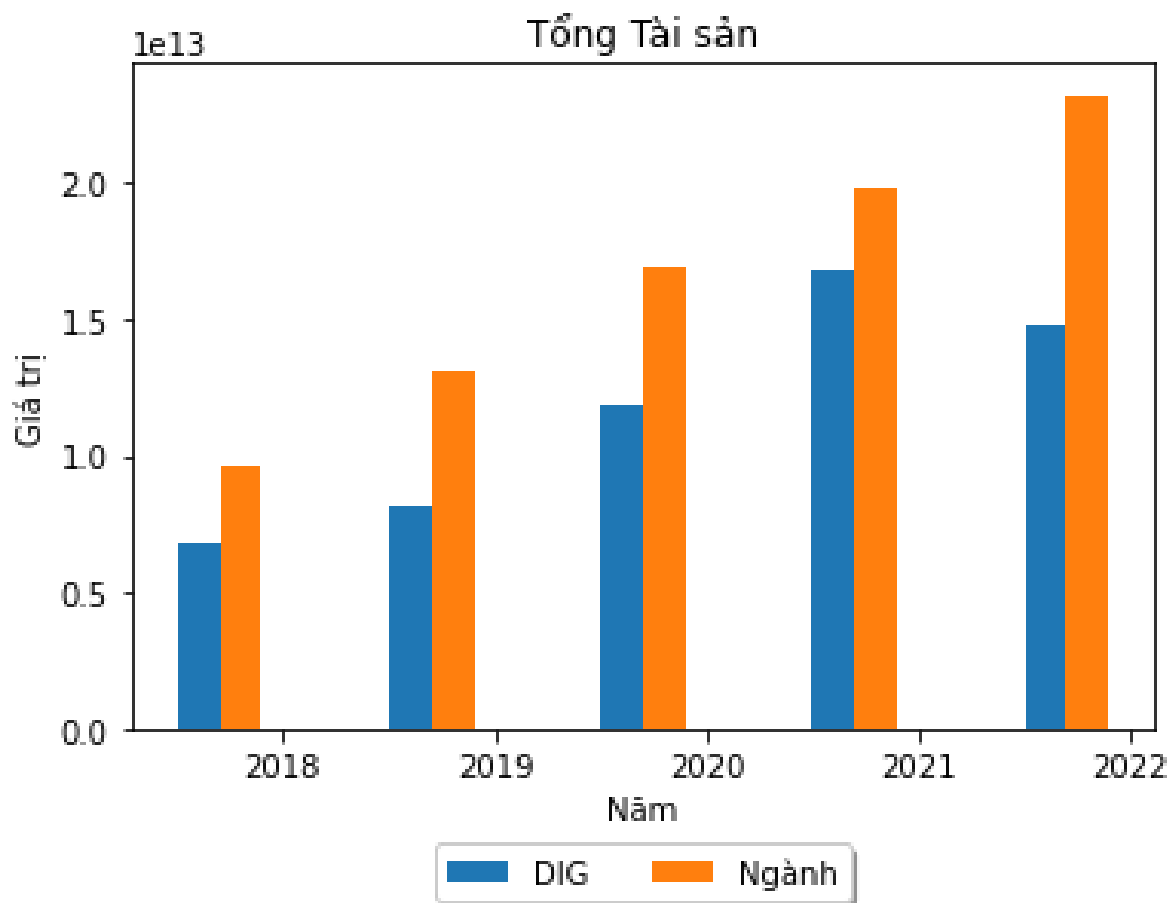
Nhân xét :

Thứ nhất, vòng quay tổng tài sản của DIG đã giảm từ 0,343233 vào năm 2018 xuống 0,120064 vào năm 2022. Điều này cho thấy cổ phiếu của công ty đang được giao dịch ít hơn so với trước đây.

Thứ hai, vòng quay hàng tồn kho của DIG cũng đã giảm, từ 0,928776 vào năm 2018 xuống 0,517835 vào năm 2022. Điều này có thể là do công ty đang nắm giữ hàng tồn kho lâu hơn trước đây.

Thứ ba, vòng quay tài sản dài hạn) của DIG đã giảm từ 1,95666 vào năm 2018 xuống 0,399174 vào năm 2022. Điều này có thể là do công ty đang đầu tư ít hơn vào tài sản dài hạn.

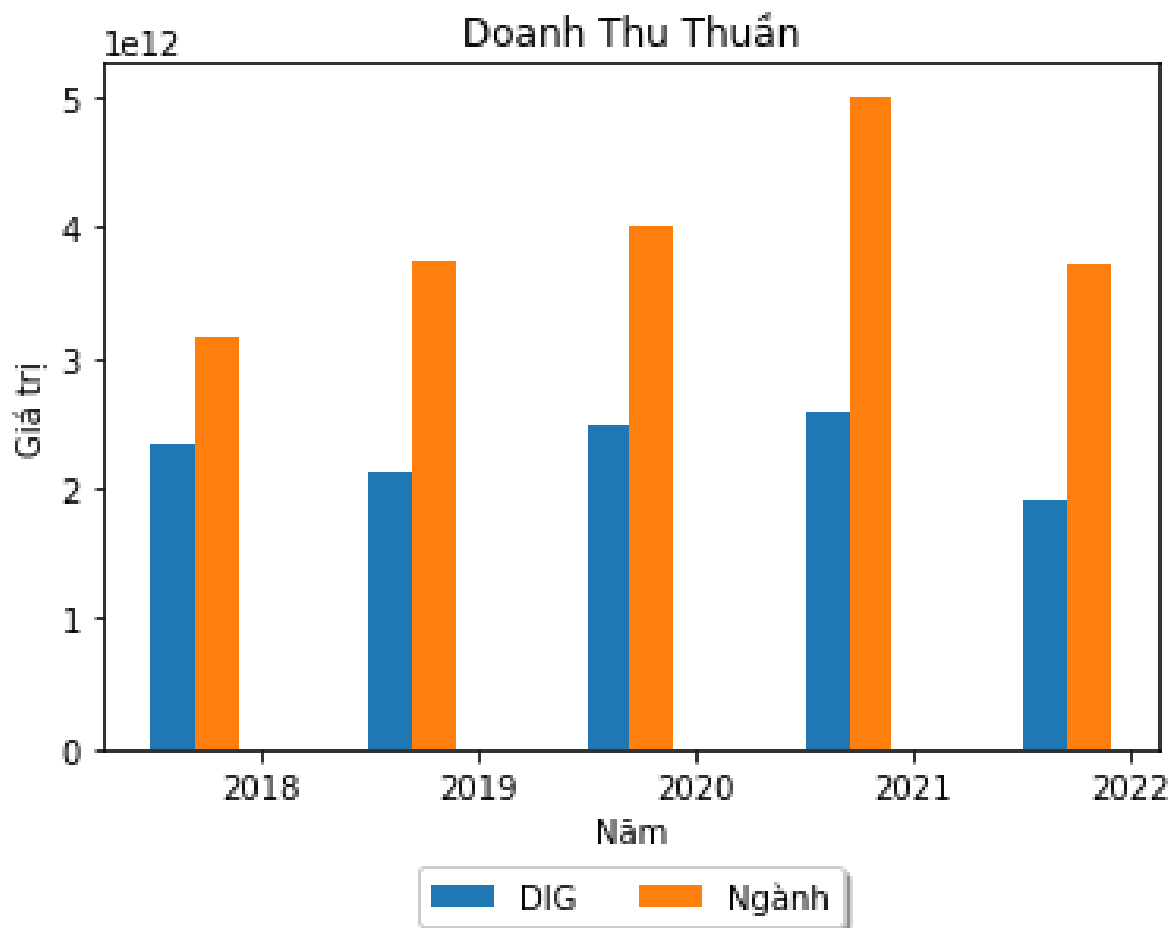
6.6. So sánh với ngành



Hình 9.1: Biểu đồ tổng tài sản của DIG so với trung bình ngành từ năm 2018 đến 2022

Nhận xét :

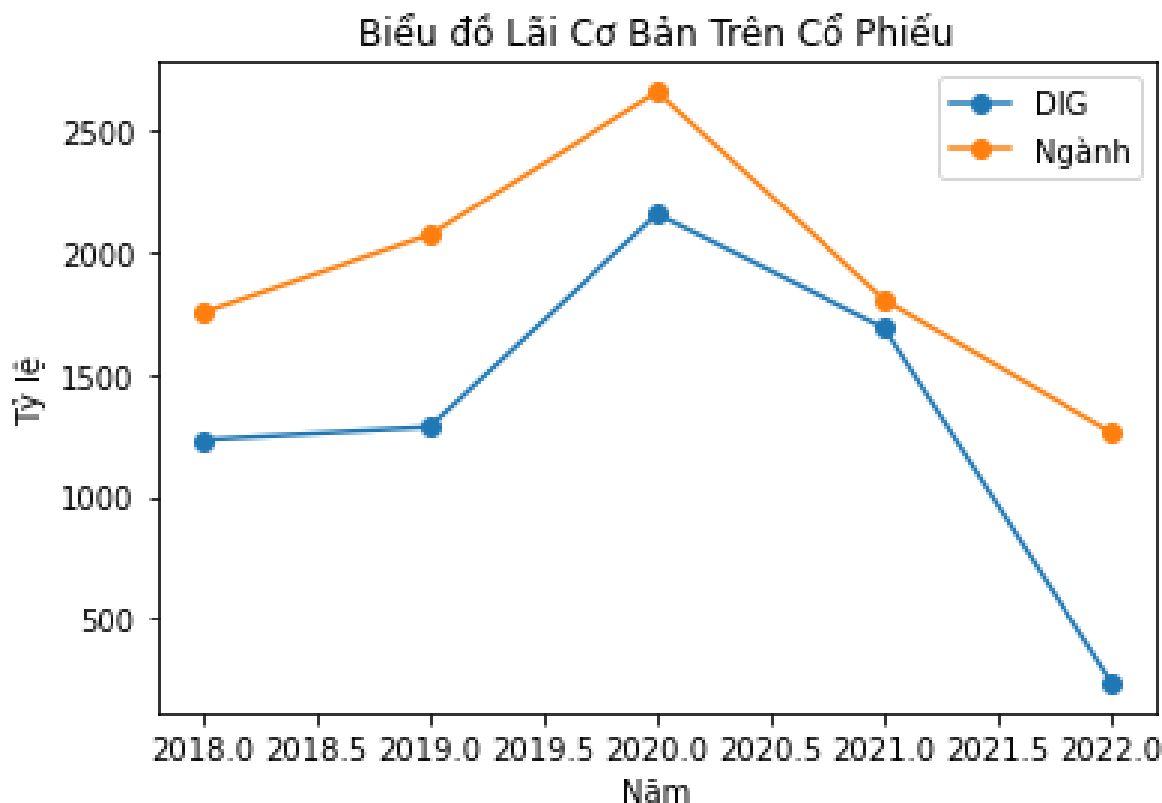
Nhìn chung tổng tài sản của DIG qua các năm nhỏ hơn trung bình của toàn ngành.



Hình 9.2: Biểu đồ doanh thu của DIG so với trung bình ngành từ năm 2018 đến 2022

Nhận xét :

Nhìn chung doanh thu của DIG qua các năm nhỏ hơn trung bình doanh thu của toàn ngành.



Hình 9.3: Biểu đồ EPS của DIG so với trung bình ngành từ năm 2018 đến 2022

Nhận xét :

Nhìn chung lãi cơ bản trên cổ phiếu của DIG qua các năm nhỏ hơn trung bình của toàn ngành. Năm 2021 lãi cơ bản trên cổ phiếu của DIG bằng với trung bình của toàn ngành, tuy nhiên năm 2022 thì lãi cơ bản trên cổ phiếu của DIG đang giảm mạnh.

7. Tổng kết

Tổng Công ty cổ phần Đầu tư Phát triển Xây dựng (DIC) là một trong những công ty bất động sản hàng đầu Việt Nam, với bề dày lịch sử hơn 40 năm. Trong giai đoạn 2018-2022, cổ phiếu DIG đã có những biến động đáng chú ý, với những điểm nhấn sau:

Tăng trưởng mạnh mẽ giai đoạn 2018-2021: Giá cổ phiếu DIG tăng trưởng mạnh mẽ trong giai đoạn này Nguyên nhân chính của sự tăng trưởng này là do DIG có kết quả kinh doanh khả quan, với doanh thu và lợi nhuận đều tăng trưởng mạnh mẽ trong giai đoạn này.

Giảm mạnh trong năm 2022: Tuy nhiên, sang năm 2022, giá cổ phiếu DIG đã giảm mạnh. Nguyên nhân chính của sự giảm giá này là do thị trường chứng khoán Việt Nam nói chung và thị trường bất động sản nói riêng đều có xu hướng giảm trong năm 2022. Ngoài ra, một số yếu tố nội tại của DIG cũng ảnh hưởng đến giá cổ phiếu, như việc công ty phải trích lập dự phòng lớn cho các khoản đầu tư thua lỗ.

Nhìn chung, mã cổ phiếu DIG trong giai đoạn 2018-2022 có sự biến động mạnh mẽ, với những giai đoạn tăng trưởng mạnh mẽ và giảm mạnh. Tóm lại, mã cổ phiếu DIG vẫn có tiềm năng tăng trưởng trong thời gian tới, nhưng nhà đầu tư cần cân nhắc kỹ lưỡng.

8. Code

Import dữ liệu

```
import plotly.express as px
import matplotlib.pyplot as plt
from datetime import date
import numpy as np
import warnings
from datetime import datetime as dt
from datetime import timedelta as delta
from datetime import datetime
from ta.trend import macd_signal
import pandas as pd
import plotly.graph_objects as go
import dash
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input, Output
app = dash.Dash(__name__)
warnings.filterwarnings("ignore")
inf = pd.read_excel(r'Price-Vol VN 2015-2023.xlsx', sheet_name='Info')
pr = pd.read_excel(r'Price-Vol VN 2015-2023.xlsx', sheet_name='Price')
vl = pd.read_excel(r'Price-Vol VN 2015-2023.xlsx', sheet_name='Volume')

# Sheet 2022
sheet2022 = pd.read_excel(r'2022-Vietnam.xlsx', index_col=0)
sheet2022 = sheet2022.iloc[7:-11]
sheet2022.columns = sheet2022.iloc[0]
sheet2022 = sheet2022.iloc[1:]

# Sheet 2021
sheet2021 = pd.read_excel(r'2018-Vietnam.xlsx', index_col=0)
sheet2021 = sheet2021.iloc[7:-11]
sheet2021.columns = sheet2021.iloc[0]
sheet2021 = sheet2021.iloc[1:]

# Clean sheet 2020
sheet2020 = pd.read_excel(r'2019-Vietnam.xlsx', index_col=0)
sheet2020 = sheet2020.iloc[7:-11]
sheet2020.columns = sheet2020.iloc[0]
sheet2020 = sheet2020.iloc[1:]

# Sheet 2019
sheet2019 = pd.read_excel(r'2020-Vietnam.xlsx', index_col=0)
sheet2019 = sheet2019.iloc[7:-11]
sheet2019.columns = sheet2019.iloc[0]
sheet2019 = sheet2019.iloc[1:]

# Sheet 2018
sheet2018 = pd.read_excel(r'2021-Vietnam.xlsx', index_col=0)
sheet2018 = sheet2018.iloc[7:-11]
sheet2018.columns = sheet2018.iloc[0]
sheet2018 = sheet2018.iloc[1:]

# Filter by 'Mã'
column_name = 'Mã'
sheet2022DIG = sheet2022[sheet2022[column_name] == 'DIG']
```

```

sheet2021DIG = sheet2021[sheet2021[column_name] == 'DIG']
sheet2020DIG = sheet2020[sheet2020[column_name] == 'DIG']
sheet2019DIG = sheet2019[sheet2019[column_name] == 'DIG']
sheet2018DIG = sheet2018[sheet2018[column_name] == 'DIG']

```

Sử lý dữ liệu ngành cấp 4 của Bất động sản

```

# Lọc DIG2022 theo ngành ICB cấp 4
filtered_2022 = sheet2022.loc[sheet2022['Ngành ICB - cấp 4'] == 'Bất động sản']

# Lọc DIG2021 theo ngành ICB cấp 4
filtered_2021 = sheet2021.loc[sheet2021['Ngành ICB - cấp 4'] == 'Bất động sản']

# Lọc DIG2020 theo ngành ICB cấp 4
filtered_2020 = sheet2020.loc[sheet2020['Ngành ICB - cấp 4'] == 'Bất động sản']

# Lọc DIG2019 theo ngành ICB cấp 4
filtered_2019 = sheet2019.loc[sheet2019['Ngành ICB - cấp 4'] == 'Bất động sản']

# Lọc DIG2018 theo ngành ICB cấp 4
filtered_2018 = sheet2018.loc[sheet2018['Ngành ICB - cấp 4'] == 'Bất động sản']

# 2022
filtered_2022 = filtered_2022.drop(filtered_2022.columns[1:10], axis=1)
filtered_2022.columns = filtered_2022.columns.str.replace(r'\n', '',
regex=True)
filtered_2022.columns = filtered_2022.columns.str.split('Hợp').str[0]
filtered_2022 = filtered_2022.dropna(axis=0)
filtered_2022.head()

# 2021
filtered_2021 = filtered_2021.drop(filtered_2021.columns[1:10], axis=1)
filtered_2021.columns = filtered_2021.columns.str.replace(r'\n', '',
regex=True)
filtered_2021.columns = filtered_2021.columns.str.split('Hợp').str[0]
filtered_2021 = filtered_2021.dropna(axis=0)
filtered_2021.head()

#2020
filtered_2020 = filtered_2020.drop(filtered_2020.columns[1:10], axis=1)
filtered_2020.columns = filtered_2020.columns.str.replace(r'\n', '',
regex=True)
filtered_2020.columns = filtered_2020.columns.str.split('Hợp').str[0]
filtered_2020 = filtered_2020.dropna(axis=0)
filtered_2020.head()

#2019
filtered_2019 = filtered_2019.drop(filtered_2019.columns[1:10], axis=1)
filtered_2019.columns = filtered_2019.columns.str.replace(r'\n', '',
regex=True)
filtered_2019.columns = filtered_2019.columns.str.split('Hợp').str[0]
filtered_2019 = filtered_2019.dropna(axis=0)
filtered_2019.head()

#2018
filtered_2018 = filtered_2018.drop(filtered_2018.columns[1:10], axis=1)

```

```

filtered_2018.columns = filtered_2018.columns.str.replace(r'\n', '',
regex=True)
filtered_2018.columns = filtered_2018.columns.str.split('Hợp').str[0]
filtered_2018 = filtered_2018.dropna(axis=0)
filtered_2018.head()

```

Sử lý dữ liệu bảng Info - Price - Volume

```

#SỬ LÝ DỮ LIỆU BẢNG PRICE
vl["Code"] = vl["Code"].fillna(pr["Code"])

def get_stock_info():
    stock_info = inf[["Symbol", "Sector"]].copy()
    stock_info.loc[:, "Symbol"] =
stock_info["Symbol"].str.split('VT:').str[1].fillna(stock_info['Symbol'])
    stock_info.loc[:, "Symbol"] = stock_info["Symbol"].str.replace('TW:',
'')
    return stock_info
a = get_stock_info()

def get_stock_price(code):
    stock_data = pr[pr["Code"].astype(str).apply(lambda x: code in x)]
    stock_data["Code"] =
stock_data["Code"].str.split("VT:").str[1].str.replace("\ (P\)",
"").fillna(
        stock_data['Code']).str.replace("\ (P\)", "", regex=True)
    stock_data["Code"] = stock_data["Code"].str.replace('TW:', '')
    stock_price = stock_data.melt(id_vars=["Name", "Code", "CURRENCY"],
var_name="Date", value_name="Close")
    stock_price = stock_price.dropna(subset=["Close"])
    return stock_price[["Date", "Code", "Close"]]
b = get_stock_price("")

def get_stock_volume(code):
    stock_data = vl[vl["Code"].astype(str).apply(lambda x: code in x)]
    stock_data["Code"] =
stock_data["Code"].str.split("VT:").str[1].str.replace("\ (VO\)",
"").fillna(
        stock_data['Code']).str.replace("\ (VO\)", "")
    stock_data["Code"] = stock_data["Code"].str.replace("\ (P\)", "")
    stock_data["Code"] = stock_data["Code"].str.replace('TW:', '')
    stock_volume = stock_data.melt(id_vars=["Name", "Code", "CURRENCY"],
var_name="Date", value_name="Volume")
    stock_volume = stock_volume.dropna(subset=["Volume"])
    stock_volume["Volume"] = pd.to_numeric(stock_volume["Volume"],
errors='coerce')
    return stock_volume[["Date", "Code", "Volume"]]
c = get_stock_volume("")

merged_vl = pd.merge(a, c, left_on="Symbol", right_on="Code")
sorted_vl = merged_vl.sort_values(by="Date", ascending=True)
merged_pr = pd.merge(a, b, left_on="Symbol", right_on="Code")
sorted_pr = merged_pr.sort_values(by="Date", ascending=True)

# Define the stock price function
def stock_pr_DIG(code):
    stock_data = pr[pr["Code"].astype(str).apply(lambda x: code in x)]

```

```

stock_price = stock_data.melt(id_vars=["Name", "Code", "CURRENCY"],
var_name="Date", value_name="Close")
stock_price = stock_price.dropna(subset=["Close"])
stock_price["Close"] = stock_price["Close"].astype(float) # Convert
"Close" column to float
return stock_price[["Date", "Close"]]
pr_DIG = stock_pr_DIG("DIG")
# Define the stock volume function
def stock_vl_DIG(code):
    stock_data = vl[vl["Code"].astype(str).apply(lambda x: code in x)]
    stock_volume = stock_data.melt(id_vars=["Name", "Code", "CURRENCY"],
var_name="Date", value_name="Volume")
    stock_volume = stock_volume.dropna(subset=["Volume"])
    stock_volume["Volume"] = stock_volume["Volume"].astype(float) #
Convert "Volume" column to float
    return stock_volume[["Date", "Volume"]]
vl_DIG = stock_vl_DIG("DIG")
if isinstance(pr_DIG, str) or isinstance(vl_DIG, str):
    print(pr_DIG)
else:
    DIG = pd.merge(pr_DIG, vl_DIG, on=["Date"], how="outer")
    print(DIG)
# Tính giá trung bình của các năm
dig_2018 = DIG[DIG['Date'].dt.year == 2018]
average_price_2018 = dig_2018['Close'].mean()
dig_2019 = DIG[DIG['Date'].dt.year == 2019]
average_price_2019 = dig_2019['Close'].mean()
dig_2020 = DIG[DIG['Date'].dt.year == 2020]
average_price_2020 = dig_2020['Close'].mean()
dig_2021 = DIG[DIG['Date'].dt.year == 2021]
average_price_2021 = dig_2021['Close'].mean()
dig_2022 = DIG[DIG['Date'].dt.year == 2022]
average_price_2022 = dig_2022['Close'].mean()
average_prices = pd.Series([average_price_2018, average_price_2019,
average_price_2020, average_price_2021, average_price_2022],
name='average_price')

```

Lấy mã cổ phiếu DIG

```

# LẤY MÃ CỔ PHIẾU DIG
#Tạo thêm Column "DIG" và "2022"
DIG2022 = sheet2022DIG.melt(
    id_vars=[
        "Mã",
        "Tên công ty",
        "Sàn",
        "Ngành ICB - cấp 1",
        "Ngành ICB - cấp 2",
        "Ngành ICB - cấp 3",
        "Ngành ICB - cấp 4",
        "Quý\nHợp nhất\nQuý: Hàng năm\nNăm: 2022\n",
        "Năm\nHợp nhất\nQuý: Hàng năm\nNăm: 2022\n",
        "Trạng thái kiểm toán\nHợp nhất\nQuý: Hàng năm\nNăm: 2022\n"
    ],
    var_name="DIG",
    value_name="2022"
)

```

```

)
#Tạo thêm Column "DIG" và "2021"
DIG2021 = sheet2021DIG.melt(
    id_vars=[
        "Mã",
        "Tên công ty",
        "Sàn",
        "Ngành ICB - cấp 1",
        "Ngành ICB - cấp 2",
        "Ngành ICB - cấp 3",
        "Ngành ICB - cấp 4",
        "Quý\Hợp nhất\Quý: Hàng năm\Năm: 2021\n",
        "Năm\Hợp nhất\Quý: Hàng năm\Năm: 2021\n",
        "Trạng thái kiểm toán\Hợp nhất\Quý: Hàng năm\Năm: 2021\n"
    ],
    var_name="DIG",
    value_name="2021"
)
#Tạo thêm Column "DIG" và "2020"
DIG2020 = sheet2020DIG.melt(
    id_vars=[
        "Mã",
        "Tên công ty",
        "Sàn",
        "Ngành ICB - cấp 1",
        "Ngành ICB - cấp 2",
        "Ngành ICB - cấp 3",
        "Ngành ICB - cấp 4",
        "Quý\Hợp nhất\Quý: Hàng năm\Năm: 2020\n",
        "Năm\Hợp nhất\Quý: Hàng năm\Năm: 2020\n",
        "Trạng thái kiểm toán\Hợp nhất\Quý: Hàng năm\Năm: 2020\n"
    ],
    var_name="DIG",
    value_name="2020"
)
#Tạo thêm Column "DIG" và "2019"
DIG2019 = sheet2019DIG.melt(
    id_vars=[
        "Mã",
        "Tên công ty",
        "Sàn",
        "Ngành ICB - cấp 1",
        "Ngành ICB - cấp 2",
        "Ngành ICB - cấp 3",
        "Ngành ICB - cấp 4",
        "Quý\Hợp nhất\Quý: Hàng năm\Năm: 2019\n",
        "Năm\Hợp nhất\Quý: Hàng năm\Năm: 2019\n",
        "Trạng thái kiểm toán\Hợp nhất\Quý: Hàng năm\Năm: 2019\n"
    ],
    var_name="DIG",
    value_name="2019"
)
#Tạo thêm Column "DIG" và "2018"
DIG2018 = sheet2018DIG.melt(
    id_vars=[
        "Mã",
        "Tên công ty",

```

```

        "Sàn",
        "Ngành ICB - cấp 1",
        "Ngành ICB - cấp 2",
        "Ngành ICB - cấp 3",
        "Ngành ICB - cấp 4",
        "Quý\NHợp nhất\Quý: Hàng năm\Năm: 2018\n",
        "Năm\NHợp nhất\Quý: Hàng năm\Năm: 2018\n",
        "Trạng thái kiểm toán\NHợp nhất\Quý: Hàng năm\Năm: 2018\n"
    ],
    var_name="DIG",
    value_name="2018"
)
# Chỉ định lại các cột giữ lại
keep2022 = ["DIG", "2022"]
DIG2022 = DIG2022[keep2022]

keep2021 = ["DIG", "2021"]
DIG2021 = DIG2021[keep2021]

keep2020 = ["DIG", "2020"]
DIG2020 = DIG2020[keep2020]

keep2019 = ["DIG", "2019"]
DIG2019 = DIG2019[keep2019]

keep2018 = ["DIG", "2018"]
DIG2018 = DIG2018[keep2018]
#Xóa các kí tự không cần thiết
DIG2022["DIG"] = DIG2022["DIG"].str.split("\\n").str[0]
DIG2021["DIG"] = DIG2021["DIG"].str.split("\\n").str[0]
DIG2020["DIG"] = DIG2020["DIG"].str.split("\\n").str[0]
DIG2019["DIG"] = DIG2019["DIG"].str.split("\\n").str[0]
DIG2018["DIG"] = DIG2018["DIG"].str.split("\\n").str[0]
# Kết hợp các bảng
DIG_table = pd.merge(DIG2018, DIG2019, on="DIG", how="outer")
DIG_table = pd.merge(DIG_table, DIG2020, on="DIG", how="outer")
DIG_table = pd.merge(DIG_table, DIG2021, on="DIG", how="outer")
DIG_table = pd.merge(DIG_table, DIG2022, on="DIG", how="outer")
# Xóa các dòng trùng nhau trong bảng DIG_table
DIG_table = DIG_table.drop_duplicates()
DIG_table.head()
# Filter the data based on specific prefixes
cdkt = DIG_table[DIG_table['DIG'].str.startswith('CĐKT.')]
kqkd = DIG_table[DIG_table['DIG'].str.startswith('KQKD.')]
lctt = DIG_table[DIG_table['DIG'].str.startswith('LCTT.')]
tm = DIG_table[DIG_table['DIG'].str.startswith('TM.')]
bctckh = DIG_table[DIG_table['DIG'].str.startswith('BCTCKH.')]

# Remove prefixes from filtered data
cdkt['DIG'] = cdkt['DIG'].str.replace('CĐKT.', '')
kqkd['DIG'] = kqkd['DIG'].str.replace('KQKD.', '')
lctt['DIG'] = lctt['DIG'].str.replace('LCTT.', '')
tm['DIG'] = tm['DIG'].str.replace('TM.', '')
bctckh['DIG'] = bctckh['DIG'].str.replace('BCTCKH.', '')

#CHUYỂN CÁC BẢNG ĐÚNG ĐỊNH DẠNG ĐỂ VẼ BIỂU ĐỒ
cdktT = cdkt.T

```



```

cdktT = cdktT.reset_index() # Reset the index to convert the row index
into a column
new_header = cdktT.iloc[0] # Store the first row as the new header
cdktT = cdktT[1:] # Remove the first row from the DataFrame
cdktT.columns = new_header # Set the new header as column headers

kqkdT = kqkd.T
kqkdT = kqkdT.reset_index() # Reset the index to convert the row index
into a column
new_header1 = kqkdT.iloc[0] # Store the first row as the new header
kqkdT = kqkdT[1:] # Remove the first row from the DataFrame
kqkdT.columns = new_header1 # Set the new header as column headers

lcttT = lctt.T
lcttT = lcttT.reset_index() # Reset the index to convert the row index
into a column
new_header2 = lcttT.iloc[0] # Store the first row as the new header
lcttT = lcttT[1:] # Remove the first row from the DataFrame
lcttT.columns = new_header2 # Set the new header as column headers
lcttT.head()

tmT = tm.T
tmT = tmT.reset_index() # Reset the index to convert the row index into a
column
new_header3 = tmT.iloc[0] # Store the first row as the new header
tmT = tmT[1:] # Remove the first row from the DataFrame
tmT.columns = new_header3 # Set the new header as column headers
tmT.head()

bctckhT = bctckh.T
bctckhT = bctckhT.reset_index() # Reset the index to convert the row
index into a column
new_header4 = bctckhT.iloc[0] # Store the first row as the new header
bctckhT = bctckhT[1:] # Remove the first row from the DataFrame
bctckhT.columns = new_header4 # Set the new header as column headers
bctckhT.head()

```

Sử lý các chỉ số để vẽ biểu đồ

```

#LẤY CÁC CHỈ SỐ CẦN VẼ
years = [2018, 2019, 2020, 2021, 2022] # Các nhóm
# Lấy giá trị cho cột tổng tài sản
tts = cdktT[cdktT.columns[20]].tolist()
# tổng tài sản bình quân
tts_binh_quan = [(tts[i-1] + tts[i]) / 2 if i > 0 and i < len(tts) else
tts[i] for i in range(len(tts))]
# vốn của chủ sở hữu
vccsh = cdktT[cdktT.columns[32]].tolist()
# vốn của chủ sở hữu bình quân
vccsh_binh_quan = [(vccsh[i-1] + vccsh[i]) / 2 if i > 0 and i < len(vccsh)
else vccsh[i] for i in range(len(vccsh))]
# Lấy giá trị cho cột hàng tồn kho
htk = cdktT[cdktT.columns[5]].tolist()
# Hàng tồn kho bình quân
htk_binh_quan = [(htk[i-1] + htk[i]) / 2 if i > 0 and i < len(htk) else
htk[i] for i in range(len(htk))]
# tài sản dài hạn

```

```

tsdh = cdktT[cdktT.columns[7]].tolist()
# tài sản dài hạn bình quân
tsdh_binh_quan = [(tsdh[i-1] + tsdh[i]) / 2 if i > 0 and i < len(tsdh)
else tsdh[i] for i in range(len(tsdh))]
# Lấy giá trị cho cột 1 (tài sản ngắn hạn)
tsnh = cdktT[cdktT.columns[1]].tolist()
# Lấy giá trị từ cột 2 đến cột 5
hainam = cdktT.iloc[4:, 2:6].sum()
# Lấy giá trị từ cột 8 đến cột 17 bỏ vị trí thứ 9
tammuoibay = cdktT.iloc[4:, [8, 10, 11, 12, 13, 14, 15, 16, 17]].sum()
# Lấy giá trị cho cột (nợ phải trả)
npt = cdktT[cdktT.columns[21]].tolist()
# Lấy giá trị cho cột (vốn của chủ sở hữu)
vccsh = cdktT[cdktT.columns[32]].tolist()
# Tính tỷ lệ nợ phải trả so với vốn chủ sở hữu (D/E)
D_E = [v1 / v2 for v1, v2 in zip(npt, vccsh)]
# Tính tỷ lệ nợ phải trả so với tài sản (D/A)
D_A = [v1 / v for v1, v in zip(npt, tts)]
# Giá trị doanh thu thuần
dtt = kqkdT[kqkdT.columns[2]].tolist()
# Giá trị Lợi nhuận thuần hdkd
lnthdkd = kqkdT[kqkdT.columns[10]].tolist()
# Giá trị lợi nhuận sau thuế TNDN (EAT)
lnst = kqkdT[kqkdT.columns[15]].tolist()
# Giá trị lợi nhuận gộp
lng = kqkdT[kqkdT.columns[3]].tolist()
# Biên lợi nhuận gộp
bienloinhuangop = [lng[i] / dtt[i] for i in range(len(dtt))]
# Biên lợi nhuận HĐKD
bienloinhuanhdkd = [lnthdkd[i] / dtt[i] for i in range(len(dtt))]
# Biên lợi nhuận ròng
bienloinhuanrong = [lnst[i] / dtt[i] for i in range(len(dtt))]
# ROE = (Lợi nhuận sau thuế / Vốn chủ sở hữu bình quân)
ROE = [lnst[i] / vccsh_binh_quan[i] for i in range(len(vccsh_binh_quan))]
# ROA = (Lợi nhuận sau thuế / Tổng tài sản bình quân)
ROA = [lnst[i] / tts_binh_quan[i] for i in range(len(tts_binh_quan))]
# ROS = (Lợi nhuận sau thuế / Doanh thu thuần)
ROS = [lnst[i] / dtt[i] for i in range(len(dtt))]
# Lấy giá trị chi phí lãi vay
cplv = cdktT[cdktT.columns[6]].tolist()
# Lấy giá trị khấu hao tài sản cố định
kh = lcttT[lcttT.columns[2]].tolist()
# Lấy giá trị lợi nhuận trước thuế (EBT)
lntt = kqkdT[kqkdT.columns[13]].tolist()
# EBIT
EBIT = [a + b for a, b in zip(cplv, lntt)]
# EBITDA
EBITDA = [c + d for c, d in zip(EBIT, kh)]
# tt = 1-t ; OCF = EBIT*(1-t) + Khấu Hao
tt = [(lnthdkd[i] - kh[i]) / EBIT[i] for i in range(len(EBIT))]
# t : thuế suất
t = [1 - tt[i] for i in range(len(tt))]
# Lấy giá trị Đầu tư mới vào TSCĐ
dtmvtcsđ = lcttT[lcttT.columns[5]].tolist()
# Lấy giá trị Thay đổi VLĐ
thaydoivlđ = lcttT[lcttT.columns[3]].tolist()
# Tính #FCFF = [EBIT(1-t) + Khấu hao] - [Đầu tư mới vào TSCĐ + Thay đổi

```

```

VLĐ]
FCFF = []
for i in range(len(years)):
    fcff_value = (EBIT[i] * (1 - t[i])) + kh[i] - (-dtmvtcsd[i] +
thaydoivld[i]) #chi phí trong bảng kí hiệu giá trị âm nên phải đổi dấu
    FCFF.append(fcff_value)
# Lấy giá trị tiền thu được các khoản đi vay
nguồnnophathanh = lcttT[lcttT.columns[12]].tolist()
# Lấy giá trị tiền trả nợ gốc vay
nguồnnophaitra = lcttT[lcttT.columns[13]].tolist()
# Net borrowing = Nguồn nợ phát hành - Nguồn nợ phải trả
vayrong = [e - f for e, f in zip(nguồnnophathanh, nguồnnophaitra)]
# Tính FCFE = FCFF + Chi phí vay * (1 - Thuế TNDN) + Nợ vay ròng mới.
FCFE = [FCFF[i] + (cplv[i] * (1 - t[i])) + vayrong[i] for i in
range(len(FCFF))]
# Lấy giá trị lưu chuyển tiền tệ ròng từ các hoạt động sản xuất kinh doanh
lcttthdkd= lcttT[lcttT.columns[4]].tolist()
# Lấy giá trị lưu chuyển tiền tệ ròng từ các hoạt động đầu tư
lcttthddt= lcttT[lcttT.columns[9]].tolist()
# Lấy giá trị lưu chuyển tiền tệ ròng từ các hoạt động tài chính
lcttthdttc= lcttT[lcttT.columns[16]].tolist()
# EPS = Lợi nhuận sau thuế - cổ tức ưu đãi/ số lượng cổ phiếu đang lưu
hành; cổ tức ưu đãi = 0
EPS = kqkdT[kqkdT.columns[18]].tolist() # Giá trị Lãi cơ bản trên cổ
phiếu EPS
#số lượng cổ phiếu đang lưu hành
slcplh = [lnst[i] / EPS[i] for i in range(len(EPS))]
#Lấy giá trị tài sản vô hình
tsvh = cdktT[cdktT.columns[12]].tolist()
# Giá trị ghi sổ trên 1 cổ phiếu = (Tổng giá trị tài sản - Giá trị tài sản
vô hình - Nợ) / Số lượng cổ phiếu lưu hành
BVPS = [(tts[i] - tsvh[i] - npt[i]) / slcplh[i] for i in
range(len(slcplh))]
# P/B = Giá cổ phiếu trung bình (average_prices) / Giá trị ghi sổ của cổ
phiếu (gtgstlcp)
P_B = [average_prices[i] / BVPS[i] for i in range(len(BVPS))]
#P/E = Giá cổ phiếu trung bình / EPS
P_E = [average_prices[i] / EPS[i] for i in range(len(EPS))]
# P/S = Giá cổ phiếu trung bình * số lượng cổ phiếu đang lưu hành / Doanh
thu thuần
P_S = [average_prices[i] * slcplh[i] / dtt[i] for i in range(len(dtt))]
# Tỷ lệ EBIT
tyleebit = [EBIT[i] / dtt[i] for i in range(len(dtt))]
# Nợ ngắn hạn
nnh = cdktT[cdktT.columns[22]].tolist()
# Tiền và các khoản tương đương tiền
tvcktdt = cdktT[cdktT.columns[2]].tolist()
# Đầu tư tài chính ngắn hạn
dttcnnh = cdktT[cdktT.columns[3]].tolist()
# Phải thu ngắn hạn
ptnh = cdktT[cdktT.columns[4]].tolist()
# Thanh toán hiện hành (CA) = Tài sản ngắn hạn / Tổng nợ ngắn hạn
CA = [tsnh[i] / nnh[i] for i in range(len(nnh))]
# Thanh toán nhanh (QR) = (Tiền và các khoản tương đương tiền + Đầu tư tài
chính ngắn hạn + Các khoản phải thu ngắn hạn) / Nợ ngắn hạn
QR = [(tvcktdt[i] + dttcnnh[i] + ptnh[i]) / nnh[i] for i in range(len(nnh))]
# Thanh toán lãi vay (TIE) = EBIT / Chi phí lãi vay

```

```

TIE = [EBIT[i] / cplv[i] for i in range(len(cplv))]
# vay nợ thuê tài chính ngắn hạn
vnh = cdktT[cdktT.columns[26]].tolist()
# vay nợ thuê tài chính dài hạn
vdh = cdktT[cdktT.columns[31]].tolist()
# Nợ = vay nợ thuê tài chính ngắn hạn + Vay nợ thuê tài chính dài hạn
no = [m + n for m, n in zip(vnh, vdh)]
# Vốn đầu tư = Vốn chủ sở hữu + Nợ - Tiền và các khoản tương đương tiền
vdt = [x + y - z for x, y, z in zip(vccsh, no, tvcktdt)]
# ROIC= Lợi nhuận sau thuế / Vốn đầu tư
ROIC = [lnst[i] / vdt[i] for i in range(len(vdt))]
# Tổng số vốn sử dụng = Tổng tài sản - Nợ ngắn hạn
tsvsd = [k - l for k, l in zip(tts, nnh)]
# ROCE = EBIT / Tổng số vốn sử dụng
ROCE = [EBIT[i] / tsvsd[i] for i in range(len(tsvsd))]
# Vòng quay tổng TS = Doanh thu thuần / tài sản bình quân
vqtts = [dtt[i] / tts_binh_quan[i] for i in range(len(tts_binh_quan))]
# Vòng quay HTK = Giá vốn bán / Hàng tồn kho bình quân = (Doanh thu thuần
- Lợi nhuận gộp) / Hàng tồn kho bình quân
vqhtk = [(dtt[i] + lng[i]) / htk_binh_quan[i] for i in
range(len(htk_binh_quan))]
# Vòng quay TSDH = Doanh thu thuần / tài sản dài hạn bình quân
vqtstdh = [dtt[i] / tsdh_binh_quan[i] for i in range(len(tsdh_binh_quan))]

# Tổng tài sản trung bình ngành Bất động sản
ttsbds2022 = filtered_2022['CĐKT. TỔNG CỘNG TÀI SẢN'].mean()
ttsbds2021 = filtered_2021['CĐKT. TỔNG CỘNG TÀI SẢN'].mean()
ttsbds2020 = filtered_2020['CĐKT. TỔNG CỘNG TÀI SẢN'].mean()
ttsbds2019 = filtered_2019['CĐKT. TỔNG CỘNG TÀI SẢN'].mean()
ttsbds2018 = filtered_2018['CĐKT. TỔNG CỘNG TÀI SẢN'].mean()
# Tổng doanh thu trung bình ngành Bất động sản
dttbds2022 = filtered_2022['KQKD. Doanh thu thuần'].mean()
dttbds2021 = filtered_2021['KQKD. Doanh thu thuần'].mean()
dttbds2020 = filtered_2020['KQKD. Doanh thu thuần'].mean()
dttbds2019 = filtered_2019['KQKD. Doanh thu thuần'].mean()
dttbds2018 = filtered_2018['KQKD. Doanh thu thuần'].mean()
#Lãi cơ bản trên cổ phiếu trung bình của ngành Bất động sản
EPSbds2022 = filtered_2022['KQKD. Lãi cơ bản trên cổ phiếu'].mean()
EPSbds2021 = filtered_2021['KQKD. Lãi cơ bản trên cổ phiếu'].mean()
EPSbds2020 = filtered_2020['KQKD. Lãi cơ bản trên cổ phiếu'].mean()
EPSbds2019 = filtered_2019['KQKD. Lãi cơ bản trên cổ phiếu'].mean()
EPSbds2018 = filtered_2018['KQKD. Lãi cơ bản trên cổ phiếu'].mean()
#Tổng tài sản trung bình ngành
ttstb = [ttsbds2018, ttsbds2019, ttsbds2020, ttsbds2021, ttsbds2022]
# Doanh thu trung bình ngành
dtttb = [dttbds2018, dttbds2019, dttbds2020, dttbds2021, dttbds2022]
# EPS trung bình ngành
EPStb = [EPSbds2018, EPSbds2019, EPSbds2020, EPSbds2021, EPSbds2022]

```

Vẽ biểu đồ Tree map theo khối lượng

```

# Vẽ Biểu đồ Treemap theo khối lượng
app = dash.Dash(__name__)

app.layout = html.Div([
    html.H1("Select Date"),
    dcc.DatePickerSingle(

```

```

        id='date-picker',
        date=datetime.now().date()
    ),
    html.Button('Select', id='select-buttonvl', n_clicks=0),
    dcc.Graph(id='treemap-graph')
])

@app.callback(
    Output('treemap-graph', 'figure'),
    [Input('select-buttonvl', 'n_clicks')],
    [dash.dependencies.State('date-picker', 'date')]
)
def update_graph(n_clicks, selected_date):
    filtered_data = merged_vl[merged_vl["Date"] == selected_date]

    # Remove rows with missing or zero values in the "Volume" column
    filtered_data = filtered_data[filtered_data["Volume"] > 0]

    # Check if there are any valid rows remaining
    if not filtered_data.empty:
        # Create the Treemap using Plotly Express
        fig = px.treemap(filtered_data, path=[px.Constant("Volume Stock
VietNam"), 'Sector', 'Symbol'], values='Volume',
                        hover_data=['Sector'], color='Volume',
                        color_continuous_scale='RdBu')

        fig.update_layout(margin=dict(t=50, l=25, r=25, b=25))
        return fig
    else:
        return {}

if __name__ == '__main__':
    app.run_server(debug=True, port=1000)

```

Vẽ biểu đồ Tree map theo giá

```

# Vẽ Biểu đồ Treemap theo giá
from datetime import datetime
app = dash.Dash(__name__)

app.layout = html.Div([
    html.H1("Select Date"),
    dcc.DatePickerSingle(
        id='date-picker',
        date=datetime.now().date()
    ),
    html.Button('Select', id='select-buttonpr', n_clicks=0),
    dcc.Graph(id='treemap-graph')
])

@app.callback(
    Output('treemap-graph', 'figure'),
    [Input('select-buttonpr', 'n_clicks')],
    [dash.dependencies.State('date-picker', 'date')]
)
def update_graph(n_clicks, selected_date):
    filtered_data = sorted_pr[sorted_pr["Date"] == selected_date]

```

```

# Remove rows with missing or zero values in the "Close" column
filtered_data = filtered_data[filtered_data["Close"] > 0]

# Check if there are any valid rows remaining
if not filtered_data.empty:
    # Create the Treemap using Plotly Express
    fig = px.treemap(filtered_data, path=[px.Constant("Price Stock
VietNam"), 'Sector', 'Symbol'], values='Close',
                    hover_data=['Sector'], color='Close',
                    color_continuous_scale='RdBu')

    fig.update_layout(margin=dict(t=50, l=25, r=25, b=25))
    return fig
else:
    return {}

if __name__ == '__main__':
    app.run_server(debug=True, port=1001)

```

Vẽ các đường chỉ báo kỹ thuật

```

# VẼ CÁC ĐƯỜNG CHỈ BÁO KỸ THUẬT
app.layout = html.Div(
    children=[
        html.H1("DIG Analysis Dashboard"),
        html.Div(
            children=[
                html.Label("Select Technical Indicators:"),
                dcc.Dropdown(
                    id="indicator-dropdown",
                    options=[
                        {"label": "SMA", "value": "SMA"},
                        {"label": "EMA", "value": "EMA"},
                        {"label": "Bbands", "value": "Bbands"},
                        {"label": "MACD", "value": "MACD"},
                        {"label": "RSI", "value": "RSI"},
                        {"label": "Stochastic", "value": "Stochastic"},
                    ],
                    multi=True,
                    value=[],
                ),
                dcc.Graph(id="stock-analysis-chart"),
            ],
        ),
    ],
)

def calculate_bollinger_bands(close_prices, window, num_std):
    rolling_mean = close_prices.rolling(window=window).mean()
    rolling_std = close_prices.rolling(window=window).std()
    upper_band = rolling_mean + num_std * rolling_std
    lower_band = rolling_mean - num_std * rolling_std
    return upper_band, rolling_mean, lower_band

```

```

def calculate_macd(close_prices, ema_short, ema_long, signal):
    ema_short = close_prices.ewm(span=ema_short, adjust=False).mean()
    ema_long = close_prices.ewm(span=ema_long, adjust=False).mean()
    macd_line = ema_short - ema_long
    signal_line = macd_line.ewm(span=signal, adjust=False).mean()
    return macd_line, signal_line

def calculate_macd_with_histogram(close_prices, ema_short, ema_long,
signal):
    ema_short = close_prices.ewm(span=ema_short, adjust=False).mean()
    ema_long = close_prices.ewm(span=ema_long, adjust=False).mean()
    macd_line = ema_short - ema_long
    signal_line = macd_line.ewm(span=signal, adjust=False).mean()
    macd_histogram = macd_line - signal_line
    return macd_line, signal_line, macd_histogram

def calculate_rsi(close_prices, window):
    delta = close_prices.diff()
    gain = delta.where(delta > 0, 0)
    loss = -delta.where(delta < 0, 0)
    avg_gain = gain.rolling(window=window).mean()
    avg_loss = loss.rolling(window=window).mean()
    rs = avg_gain / avg_loss
    rsi = 100 - (100 / (1 + rs))
    return rsi

def calculate_stochastic(close_prices, high_prices, low_prices, window):
    lowest_low = low_prices.rolling(window=window).min()
    highest_high = high_prices.rolling(window=window).max()
    stochastic_k = (close_prices - lowest_low) / (highest_high -
lowest_low) * 100
    stochastic_d = stochastic_k.rolling(window=3).mean() # You can adjust
the window for %K calculation
    return stochastic_k, stochastic_d

@app.callback(
    Output("stock-analysis-chart", "figure"),
    Input("indicator-dropdown", "value"),
)
def update_chart(indicators):
    fig = go.Figure()

    # Add stock price trace
    fig.add_trace(
        go.Scatter(
            x=pr_DIG["Date"],
            y=pr_DIG["Close"],
            name="Stock Price",
            line=dict(color="blue"),
        )
    )

```

```

# Add stock volume bar chart
fig.add_trace(
    go.Bar(
        x=vl_DIG["Date"],
        y=vl_DIG["Volume"],
        name="Volume",
        marker=dict(color="red"),
    )
)

# Add selected technical indicators
for indicator in indicators:
    if indicator == "SMA":
        # Add Simple Moving Average (SMA)
        sma_20 = pr_DIG["Close"].rolling(window=20).mean()
        fig.add_trace(
            go.Scatter(
                x=pr_DIG["Date"],
                y=sma_20,
                name="SMA (20)",
                line=dict(color="orange"),
            )
        )
    elif indicator == "EMA":
        # Add Exponential Moving Average (EMA)
        ema_12 = pr_DIG["Close"].ewm(span=12, adjust=False).mean()
        fig.add_trace(
            go.Scatter(
                x=pr_DIG["Date"],
                y=ema_12,
                name="EMA (12)",
                line=dict(color="green"),
            )
        )
    elif indicator == "Bbands":
        # Add Bollinger Bands
        upper_band, middle_band, lower_band =
calculate_bollinger_bands(pr_DIG["Close"], window=20, num_std=2)
        fig.add_trace(
            go.Scatter(
                x=pr_DIG["Date"],
                y=upper_band,
                name="Upper Bollinger Band",
                line=dict(color="purple"),
            )
        )
        fig.add_trace(
            go.Scatter(
                x=pr_DIG["Date"],
                y=middle_band,
                name="Middle Bollinger Band",
                line=dict(color="red"),
            )
        )
        fig.add_trace(
            go.Scatter(
                x=pr_DIG["Date"],

```



```

        y=lower_band,
        name="Lower Bollinger Band",
        line=dict(color="purple"),
    )
)
elif indicator == "MACD":
    # Add MACD
    macd_line, signal_line, macd_histogram =
calculate_macd_with_histogram(pr_DIG["Close"], ema_short=12,
ema_long=26, signal=9)
    fig.add_trace(
        go.Scatter(
            x=pr_DIG["Date"],
            y=macd_line,
            name="MACD Line",
            line=dict(color="red"),
        )
    )
    fig.add_trace(
        go.Scatter(
            x=pr_DIG["Date"],
            y=signal_line,
            name="Signal Line",
            line=dict(color="blue"),
        )
    )
    fig.add_trace(
        go.Bar(
            x=pr_DIG["Date"],
            y=macd_histogram,
            name="MACD Histogram",
        )
    )
elif indicator == "RSI":
    # Add RSI
    rsi = calculate_rsi(pr_DIG["Close"], window=14)
    fig.add_trace(
        go.Scatter(
            x=pr_DIG["Date"],
            y=rsi,
            name="RSI",
            line=dict(color="green"),
        )
    )
elif indicator == "Stochastic":
    # Add Stochastic
    stochastic_k, stochastic_d =
calculate_stochastic(pr_DIG["Close"], pr_DIG["Close"], pr_DIG["Close"],
window=14)
    fig.add_trace(
        go.Scatter(
            x=pr_DIG["Date"],
            y=stochastic_k,
            name="%K",
            line=dict(color="orange"),
        )
    )

```

```

    )

    fig.update_layout(
        title="DIG Analysis",
        xaxis=dict(title="Date"),
        yaxis=dict(title="Value"),
    )

    return fig

if __name__ == "__main__":
    app.run_server(debug=True, port=1002)

```

Vẽ các bảng

```

#TẠO CÁC BẢNG
# Read data from the DIG_table DataFrame
df = pd.DataFrame(DIG_table)
# Create a list of options for the dropdown menu
options = [
    {"label": "Cân Đối Kế Toán", "value": "cdkt"},
    {"label": "Kết Quả Kinh Doanh", "value": "kqkd"},
    {"label": "Lưu Chuyển Tiền Tệ", "value": "lctt"},
    {"label": "Thuyết Minh", "value": "tm"},
    {"label": "Báo Cáo Tài Chính Kế Hoạch", "value": "bctckh"},
]

# Create the Dash application
app = dash.Dash(__name__)

# Create the Dash application layout
app.layout = html.Div(
    children=[
        html.H1("Bảng DIG"),
        dcc.Dropdown(
            id="select-option",
            options=options,
            value=options[0]["value"], # Default value
        ),
        html.Br(),
        dcc.Graph(id="dig-table"),
    ]
)

# Define the callback to update the table based on the selected value
@app.callback(
    dash.dependencies.Output("dig-table", "figure"),
    [dash.dependencies.Input("select-option", "value")]
)
def update_table(selected_value):
    if selected_value == "cdkt":
        filtered_df = cdkt
    elif selected_value == "kqkd":
        filtered_df = kqkd
    elif selected_value == "lctt":

```

```

        filtered_df = lctt
    elif selected_value == "tm":
        filtered_df = tm
    elif selected_value == "bctckh":
        filtered_df = bctckh
    else:
        filtered_df = df

    table_data = [
        {
            "type": "table",
            "header": {
                "values": filtered_df.columns.tolist(),
                "align": "center",
                "fill": {"color": "grey"},
                "font": {"size": 14, "color": "white"},
            },
            "cells": {
                "values": filtered_df.values.T.tolist(),
                "align": "center",
                "font": {"size": 12, "color": ["black"] *
len(filtered_df.columns)},
            },
        }
    ]

    return {
        "data": table_data,
        "layout": {
            "margin": {"l": 20, "r": 20, "t": 50, "b": 50},
            "autosize": True,
        },
    }

# Run the Dash application
if __name__ == "__main__":
    app.run_server(debug=True, port=1004)

```

Vẽ biểu đồ tài sản ngắn hạn và dài hạn

```

# Tạo biểu đồ TÀI SẢN NGẮN HẠN VÀ TÀI SẢN DÀI HẠN
fig = go.Figure()

fig.add_trace(go.Bar(x=years, y=tsnh, name='TÀI SẢN NGẮN HẠN',
                    marker=dict(color='orange'), width=0.4))
fig.add_trace(go.Bar(x=years, y=tsdh, name='TÀI SẢN DÀI HẠN',
                    marker=dict(color='blue'), width=0.4))
fig.add_trace(go.Scatter(x=years, y=tts, name='TỔNG TÀI SẢN',
                        mode='lines+markers', line=dict(color='green'),
                        marker=dict(color='green'))))
# Đặt nhãn cho phần trăm tài sản ngắn hạn
for i, v1, v2 in zip(range(len(years)), tsnh, tsdh):
    if v1 + v2 != 0:
        y_position = v1/2
        fig.add_annotation(x=years[i], y=y_position, text=f'{100 * v1 /
(v1 + v2):.2f}%',
                        showarrow=False, font=dict(size=12))

```

```
# Đặt nhãn cho phần trăm tài sản dài hạn
for i, v1, v2 in zip(range(len(years)), tsnh, tsdh):
    if v1 + v2 != 0:
        y_position = v1+v2/2
        fig.add_annotation(x=years[i], y=y_position, text=f'{100 * v2 / (v1 + v2):.2f}%',
                           showarrow=False, font=dict(size=12))

# Đặt tiêu đề và chú thích
fig.update_layout(title='Biểu đồ Tài Sản của DIG theo các năm',
barmode='stack')

# Hiển thị biểu đồ
fig.show()
```

Vẽ biểu đồ tài sản ngắn hạn và dài hạn 2022

```
# TÀI SẢN NGẮN HẠN VÀ TÀI SẢN DÀI HẠN CHI TIẾT
# Tạo biểu đồ hình tròn thứ nhất
fig1 = go.Figure(data=[go.Pie(labels=hainam.index, values=hainam)])
fig1.update_layout(title='TÀI SẢN NGẮN HẠN 2022')

# Hiển thị biểu đồ thứ nhất
fig1.show()

# Loại bỏ các giá trị bằng 0
nonzero_values = []
nonzero_labels = []
for label, value in zip(tammuoibay.index, tammuoibay):
    if value != 0:
        nonzero_values.append(value)
        nonzero_labels.append(label)

# Tạo biểu đồ hình tròn thứ hai
fig2 = go.Figure(data=[go.Pie(labels=nonzero_labels,
values=nonzero_values)])
fig2.update_layout(title='TÀI SẢN DÀI HẠN 2022')

# Hiển thị biểu đồ thứ hai
fig2.show()
```

Vẽ biểu đồ nguồn vốn và các chỉ số D/E, D/A

```
# BIỂU ĐỒ NGUỒN VỐN
fig = go.Figure()

# Vẽ cột 1 (nợ phải trả)
fig.add_trace(go.Bar(x=years, y=npt, name='NỢ PHẢI TRẢ',
marker=dict(color='#B19CD9'), width=0.4))

# Vẽ cột 2 (vốn của chủ sở hữu)
fig.add_trace(go.Bar(x=years, y=vccsh, name='VỐN CỦA CHỦ SỞ HỮU',
marker=dict(color='#6F366F'), width=0.4))

# Vẽ dòng line cho D/E (tỷ lệ nợ phải trả so với vốn chủ sở hữu)
fig.add_trace(go.Scatter(x=years, y=D_E, name='TỶ LỆ D/E',
```

```

mode='lines+markers', line=dict(color='green'),
marker=dict(color='green'), yaxis='y2'))

# Vẽ dòng line cho D/A (tỷ lệ nợ phải trả so với tài sản)
fig.add_trace(go.Scatter(x=years, y=D_A, name='TỶ LỆ D/A',
mode='lines+markers', line=dict(color='blue'),
marker=dict(color='blue'), yaxis='y2'))

# Đặt nhãn cho phần trăm nợ phải trả
for i, v1, v2 in zip(range(len(years)), npt, vccsh):
    if v1 + v2 != 0:
        y_position = v1 / 2
        fig.add_annotation(x=years[i], y=y_position, text=f'{100 * v1 /
(v1 + v2) :.2f}%',
showarrow=False, font=dict(size=12))

# Đặt nhãn cho phần trăm vốn của chủ sở hữu
for i, v1, v2 in zip(range(len(years)), npt, vccsh):
    if v1 + v2 != 0:
        y_position = v1 + v2 / 2
        fig.add_annotation(x=years[i], y=y_position, text=f'{100 * v2 /
(v1 + v2) :.2f}%',
showarrow=False, font=dict(size=12))

# Đặt tiêu đề và chú thích
fig.update_layout(title='BIỂU ĐỒ BIẾN ĐỘNG NGUỒN VỐN', barmode='stack')

# Đặt nhãn cho trục y của cột
fig.update_layout(yaxis=dict(title='Giá trị'))

# Tạo trục phụ cho dòng line D/E và D/A
fig.update_layout(yaxis2=dict(overlaping='y', side='right'))

# Hiện thị biểu đồ
fig.show()

```

Vẽ biểu đồ kết quả kinh doanh

```

# Biểu đồ kết quả kinh doanh

# Độ rộng của mỗi cột
bar_width = 0.25
# Vị trí của các nhóm trên trục x
x = range(len(years))
# Tạo biểu đồ
fig, ax = plt.subplots()
rects1 = ax.bar(x, dtt, bar_width, label='Doanh Thu Thuần')
rects2 = ax.bar([i + bar_width for i in x], lnthdkd, bar_width, label='LN
Thuần HỖKD')
rects3 = ax.bar([i + 2 * bar_width for i in x], lnst, bar_width, label='LN
Sau Thuế TNDN')

# Đặt nhãn cho các nhóm trên trục x
ax.set_xticks([i + bar_width for i in x])
ax.set_xticklabels(years)

# Tạo trục y phụ

```

```

ax2 = ax.twinx()

# Vẽ ba đường line trên trục y phụ
line1 = ax2.plot(x, bienloinhuangop, marker='o', linestyle='--',
label='Biên LN Gộp')
line2 = ax2.plot(x, bienloinhuanhkd, marker='o', linestyle='--',
label='Biên LN Từ H&KD')
line3 = ax2.plot(x, bienloinhuanrong, marker='o', linestyle='--',
label='Biên LN Ròng')

# Đặt tiêu đề
ax.set_title('Biểu đồ kết quả kinh doanh của DIG ')

#Hiển thị chú thích của các dòng
ax2.legend(loc='upper center', bbox_to_anchor=(0.5, -0.1), fancybox=True,
shadow=True, ncol=3)
# Hiển thị chú thích của các cột
ax.legend(loc='upper center', bbox_to_anchor=(0.5, -0.25), fancybox=True,
shadow=True, ncol=3)

# Hiển thị biểu đồ
plt.show()

```

Vẽ các biểu đồ đường ROS, ROE, ROA

```

# BIỂU ĐỒ ĐƯỜNG ROE, ROA, ROS
# Vẽ biểu đồ đường
plt.plot(years, ROA, marker='o', linestyle='--', label='ROA')
plt.plot(years, ROE, marker='o', linestyle='--', label='ROE')
plt.plot(years, ROS, marker='o', linestyle='--', label='ROS')
# Đặt tiêu đề và nhãn cho trục x và y
plt.title('Biểu đồ ROA, ROE, ROS')
plt.xlabel('Năm')
plt.ylabel('Tỷ lệ')

# Hiển thị chú thích
plt.legend()

# Hiển thị biểu đồ
plt.show()

```

Vẽ biểu đồ lợi nhuận

```

#BIỂU ĐỒ LỢI NHUẬN
# Tạo mảng vị trí cột
bar_width = 0.2
bar_positions_lnst = np.arange(len(years))
bar_positions_lntt = bar_positions_lnst + bar_width
bar_positions_ebit = bar_positions_lnst + 2 * bar_width
bar_positions_ebitda = bar_positions_lnst + 3 * bar_width

# Vẽ biểu đồ cột
plt.bar(bar_positions_lnst, lnst, width=bar_width, label='Lợi nhuận sau
thuế TNDN (EAT)')
plt.bar(bar_positions_lntt, lntt, width=bar_width, label='Lợi nhuận trước

```

```
thuế (EBT)')
plt.bar(bar_positions_ebit, EBIT, width=bar_width, label='EBIT')
plt.bar(bar_positions_ebitda, EBITDA, width=bar_width, label='EBITDA')

# Đặt tiêu đề và nhãn trục
plt.title('Biểu đồ EAT, EBT, EBIT và EBITDA')
plt.xlabel('Năm')
plt.ylabel('Giá trị')

# Đặt nhãn trục x
plt.xticks(bar_positions_lntt + 2 * bar_width, years)

# Hiển thị chú thích
plt.legend(loc='upper center', bbox_to_anchor=(0.5, -0.15), fancybox=True,
shadow=True, ncol=3)

# Hiển thị biểu đồ
plt.show()
```

Vẽ biểu đồ lưu chuyển tiền tệ

```
#BIỂU ĐỒ LƯU CHUYỂN TIỀN TỆ
bar_width = 0.3 # Độ rộng của cột
plt.figure(figsize=(10, 6))
# Vẽ biểu đồ cột cho các giá trị lưu chuyển tiền tệ ròng
plt.bar(years, lcttthdkd, width=bar_width, label='LCTT Sản xuất kinh
doanh')
plt.bar(years, lcttthddt, width=bar_width, label='LCTT Đầu tư', alpha=0.7)
plt.bar(years, lcttthdtc, width=bar_width, label='LCTT Tài chính',
alpha=0.7)

# Vẽ đường line cho FCFF và FCFE trên cùng một cột Y
plt.plot(years, FCFF, marker='o', linestyle='--', label='FCFF',
color='red')
plt.plot(years, FCFE, marker='o', linestyle='--', label='FCFE',
color='blue')

plt.xlabel('Năm')
plt.ylabel('Giá trị')
plt.title('Biểu đồ lưu chuyển tiền tệ')

# Hiển thị chú thích LCTT và các đường line
plt.legend(loc='upper center', bbox_to_anchor=(0.5, -0.1), fancybox=True,
shadow=True, ncol=3)

plt.tight_layout()
plt.show()
```

Vẽ biểu đồ tài sản so với trung bình ngành

```
#BIỂU ĐỒ TÀI SẢN CỦA NGÀNH
# Tạo mảng vị trí cột
bar_width = 0.2
bar_DIG = np.arange(len(years))
bar_BDS = bar_DIG + bar_width

# Vẽ biểu đồ cột
```

```
plt.bar(bar_DIG, tts, width=bar_width, label='DIG')
plt.bar(bar_BDS, ttstb, width=bar_width, label='Ngành')

# Đặt tiêu đề và nhãn trục
plt.title('Tổng Tài sản')
plt.xlabel('Năm')
plt.ylabel('Giá trị')

# Đặt nhãn trục x
plt.xticks(bar_DIG + 2 * bar_width, years)

# Hiện thị chú thích
plt.legend(loc='upper center', bbox_to_anchor=(0.5, -0.15), fancybox=True,
shadow=True, ncol=3)

# Hiện thị biểu đồ
plt.show()
```

Vẽ biểu đồ doanh thu so với trung bình ngành

```
#BIỂU ĐỒ Doanh thu CỦA NGÀNH
# Tạo mảng vị trí cột
bar_width = 0.2
bar_DIG = np.arange(len(years))
bar_BDS = bar_DIG+ bar_width

# Vẽ biểu đồ cột
plt.bar(bar_DIG, dtt, width=bar_width, label='DIG')
plt.bar(bar_BDS, dtstb, width=bar_width, label='Ngành')

# Đặt tiêu đề và nhãn trục
plt.title('Doanh Thu Thuần')
plt.xlabel('Năm')
plt.ylabel('Giá trị')

# Đặt nhãn trục x
plt.xticks(bar_DIG + 2 * bar_width, years)

# Hiện thị chú thích
plt.legend(loc='upper center', bbox_to_anchor=(0.5, -0.15), fancybox=True,
shadow=True, ncol=3)

# Hiện thị biểu đồ
plt.show()
```

Vẽ biểu đồ EPS so với trung bình ngành

```
# BIỂU ĐỒ ĐƯỜNG EPS
# Vẽ biểu đồ đường
plt.plot(years, EPS, marker='o', linestyle='-', label='DIG')
plt.plot(years, EPStb, marker='o', linestyle='-', label='Ngành')

# Đặt tiêu đề và nhãn cho trục x và y
plt.title('Biểu đồ Lãi Cơ Bản Trên Cổ Phiếu')
plt.xlabel('Năm')
plt.ylabel('Tỷ lệ')
```



```
# Hiển thị chú thích
plt.legend()

# Hiển thị biểu đồ
plt.show()
```

Tạo các bảng chỉ số tài chính

```
#Tạo bảng Định giá
dinhgia = pd.DataFrame({
    "Năm": years,
    "EPS cơ bản": EPS,
    "P/E": P_E,
    "P/B": P_B,
    "P/S": P_S
})
dinhgia['Năm'] = dinhgia['Năm'].astype(str).replace('\.0', '', regex=True)

dinhgiaT = dinhgia.T
dinhgiaT = dinhgiaT.reset_index() # Reset the index to convert the row
index into a column
dinhgia2 = dinhgiaT.iloc[0] # Store the first row as the new header
dinhgiaT = dinhgiaT[1:] # Remove the first row from the DataFrame
dinhgiaT.columns = dinhgia2 # Set the new header as column headers
print(dinhgiaT)

#Tạo bảng khả năng sinh lợi
kns1 = pd.DataFrame({
    "Năm": years,
    "Tỷ lệ lãi ròng (%)": bienloinhuanrong,
    "Tỷ lệ lãi gộp (%) ": bienloinhuangop,
    "Tỷ lệ EBIT (%) ": tyleebit,
    "Tỷ lệ lãi từ HĐKD (%) ": bienloinhuanhdkd
})
kns1['Năm'] = kns1['Năm'].astype(str).replace('\.0', '', regex=True)

kns1T = kns1.T
kns1T = kns1T.reset_index() # Reset the index to convert the row index
into a column
kns12 = kns1T.iloc[0] # Store the first row as the new header
kns1T = kns1T[1:] # Remove the first row from the DataFrame
kns1T.columns = kns12 # Set the new header as column headers
print(kns1T)

#tạo bảng sức mạnh tài chính
smtc = pd.DataFrame({
    "Năm": years,
    "Thanh toán hiện hành": CA,
    "Thanh toán nhanh": QR,
    "Thanh toán lãi vay": TIE,
    "D/E ": D_E,
    "D/A": D_A
})
smtc['Năm'] = smtc['Năm'].astype(str).replace('\.0', '', regex=True)

smtcT = smtc.T
smtcT = smtcT.reset_index() # Reset the index to convert the row index
```

```

into a column
smtc2 = smtcT.iloc[0] # Store the first row as the new header
smtcT = smtcT[1:] # Remove the first row from the DataFrame
smtcT.columns = smtc2 # Set the new header as column headers
print(smtcT)

#tạo bảng hiệu quả quản lý
hqql = pd.DataFrame({
    "Năm": years,
    "ROA": ROA,
    "ROE ": ROE,
    "ROIC ": ROIC,
    "ROCE ": ROCE
})
hqql['Năm'] = hqql['Năm'].astype(str).replace('\.0', '', regex=True)

hqqlT = hqql.T
hqqlT = hqqlT.reset_index() # Reset the index to convert the row index
into a column
hqql2 = hqqlT.iloc[0] # Store the first row as the new header
hqqlT = hqqlT[1:] # Remove the first row from the DataFrame
hqqlT.columns = hqql2 # Set the new header as column headers
print(hqqlT)

#Tạo bảng khả năng hoạt động
knhd = pd.DataFrame({
    "Năm": years,
    "Vòng quay tổng TS": vqtts,
    "Vòng quay HTK ": vqhtk,
    "Vòng quay TSDH ": vqtsdh
})
knhd['Năm'] = knhd['Năm'].astype(str).replace('\.0', '', regex=True)

knhdT = knhd.T
knhdT = knhdT.reset_index() # Reset the index to convert the row index
into a column
knhd2 = knhdT.iloc[0] # Store the first row as the new header
knhdT = knhdT[1:] # Remove the first row from the DataFrame
knhdT.columns = knhd2 # Set the new header as column headers
print(knhdT)

```