

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KINH TẾ - LUẬT
KHOA TÀI CHÍNH - NGÂN HÀNG



BÁO CÁO CUỐI KỲ
MÔN: GÓI PHẦN MỀM ỨNG DỤNG TRONG TÀI CHÍNH 1
ĐỀ TÀI: BÁO CÁO PHÂN TÍCH CÔNG TY CỔ PHẦN VINHOMES
(VHM)

Giảng viên: ThS. Ngô Phú Thanh

Sinh viên thực hiện: Trần Thị Bình Nhi

Mã số sinh viên: K214140948

Lớp: K21414A

TP Hồ Chí Minh, ngày 13 tháng 1 năm 2024

Báo cáo phân tích

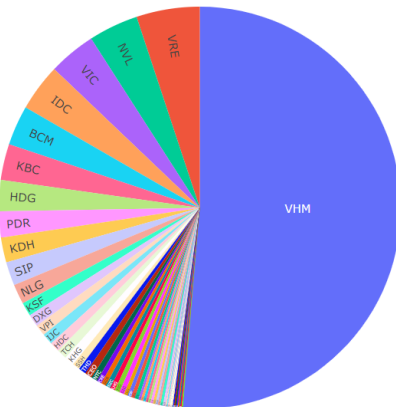
Công ty Cổ phần VINHOMES – VHM



Biểu đồ giá của VHM, KDH, NVL, PDR và NLG



Tỷ trọng doanh thu của các công ty trong ngành bất động sản (2022)



Điểm nổi bật của VHM:

- VHM ghi nhận doanh thu năm 2022 đạt, biên lợi nhuận ròng tăng so với năm 2021
- Hệ số thanh toán lãi vay năm 2022 đạt , cao hơn đáng kể so với một số doanh nghiệp lớn trong ngành.
- Công ty sử dụng hiệu quả đồng vốn của cổ đông, tỷ số lợi nhuận ròng trên vốn chủ sở hữu (ROE) cao hơn các cổ phiếu cùng ngành.
- Nợ vay ít, gánh nặng lãi vay của VHM thấp.
- Doanh thu chiếm tỷ trọng lớn trong ngành Bất động sản.

Giới thiệu về CTCP Vinhomes (VHM):

Công ty Cổ phần Vinhomes là một trong những công ty giữ vị thế dẫn đầu ngành Bất động sản, chuyên hoạt động trong lĩnh vực phát triển, chuyển nhượng, vận hành và khai thác bất động sản nhà ở phức hợp. Vinhomes bao gồm bốn dòng sản phẩm gồm các sản phẩm nhà ở thương mại Vinhomes Diamond, Vinhomes Ruby, Vinhomes Sapphire, và nhà ở xã hội Happy Home; các dự án hiện đều ở vị trí đắc địa tại các tỉnh trọng điểm. Hiện nay, công ty vẫn không ngừng phát triển và còn khá nhiều tiềm năng tăng trưởng.

Lịch sử hình thành:

Thời điểm	Sự kiện
2008	Tiền thân Công ty Cổ phần Đô thị BIDV-PP được thành lập với vốn điều lệ ban đầu là 300 tỷ đồng.
2009	<ul style="list-style-type: none">• Đổi tên thành Công ty Cổ phần Phát triển Đô thị Nam Hà Nội.• Tăng vốn điều lệ lên 500 tỷ đồng.
2011	<ul style="list-style-type: none">• Trở thành công ty đại chúng theo Công văn số 3585/UBCK-QLPH của Ủy ban chứng khoán Nhà nước.• Công ty nhận được Quyết định về việc chấp thuận đăng ký giao dịch cổ phiếu tại Sở Giao dịch Chứng khoán Hà Nội (UPCoM) với mã chứng khoán là NHN.
2018	<ul style="list-style-type: none">• Đổi tên thành Công ty Cổ phần Vinhomes.• Niêm yết trên Sở Giao dịch Chứng khoán Thành phố Hồ Chí Minh (HOSE).• Ngày 17/05/2018: là ngày giao dịch đầu tiên trên sàn HOSE với giá tham chiếu là 92,100 đ/CP.
2019	<p>Công bố quy hoạch lại các dòng sản phẩm thành 3 cấp độ: Sapphire, Ruby và Diamond.</p> <p>Công bố kế hoạch mở rộng đầu tư sang bất động sản khu công nghiệp tại các thành phố lớn, hạ tầng tốt như Hải Phòng, Quảng Ninh và phát triển hạng mục văn phòng trong các khu đô thị Vinhomes</p>
2021	<ul style="list-style-type: none">• Thủ tướng Chính phủ phê duyệt chủ trương đầu tư dự án khu đô thị sinh thái Dream City• Thủ tướng Chính phủ ban hành Quyết định 529/QĐ-TTg về việc phê duyệt chủ trương đầu tư Dự án Khu đô thị phức hợp Hạ Long Xanh.
2022	Mở bán dự án Vinhomes The Empire và Vinhomes The Crown

Khuyến khích lựa chọn Vinhomes (VHM VN, HOSE) vì:

- **VHM là nhà phát triển bất động sản (BDS) hàng đầu** với tổng quỹ đất tính đến cuối năm 2021 là khoảng 16.800ha với 90% trong số đó chưa được phát triển. VHM có đóng góp lớn trên thị trường BDS Việt Nam với **thị phần trung bình giai đoạn 2019-Q3/22 là 44%** ở phân khúc căn hộ cao cấp, 43% ở phân khúc căn hộ trung cấp và 31% ở phân khúc bất động sản liền thổ, theo CBRE.

- Trong bối cảnh ngân hàng thắt chặt cho vay vào lĩnh vực BDS và giám sát chặt chẽ hơn việc phát hành trái phiếu doanh nghiệp, thanh khoản không phải là vấn đề đáng lo ngại đối với VHM. **Tiền và các khoản tương đương tiền tăng gấp 3 lần từ tháng 12/2021** cho thấy dòng tiền được hỗ trợ bởi các dự án mở bán trong thời gian gần đây. Mặc dù VHM đã tiếp cận thị trường vốn quốc tế, nhưng chúng tôi cho rằng VHM vẫn tiếp tục tập trung vay bằng VND (chiếm 100% tổng dư nợ, điều này cũng hạn chế tác động từ biến động tỷ giá gần đây) và **duy trì đòn bẩy thấp với tỷ lệ nợ ròng/vốn chủ sở hữu là 24%** tại cuối năm 2022 để giảm thiểu những rủi ro tiềm ẩn.

- So với các công ty cùng ngành, **VHM nằm trong top sở hữu bằng cân đối lành mạnh** với tỷ lệ D/E duy trì ở quanh mức 0,2 lần trong vòng 2 năm trở lại đây, thấp hơn nhiều so với mức trung bình ngành. Bên cạnh đó, VHM đã cho thấy **hệ số thanh toán lãi vay đáng chú ý ở mức 23,9 lần** trong năm 2022, cao hơn đáng kể so với mức trung bình ngành. Những yếu tố này giúp **giảm thiểu các rủi ro liên quan đến thanh khoản và vỡ nợ**.

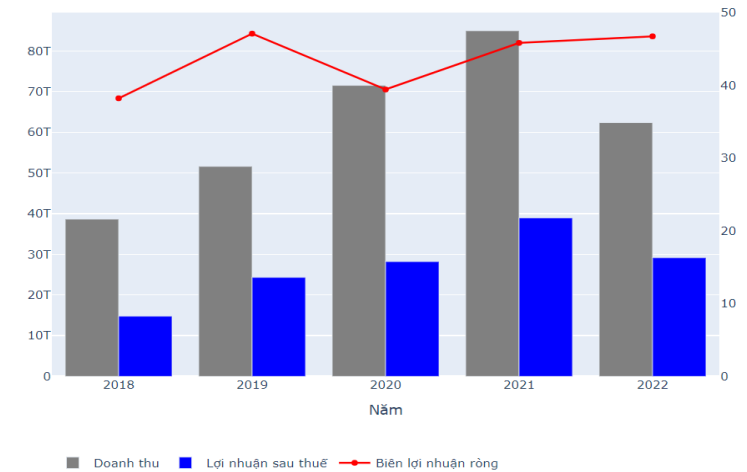
Tình hình kinh doanh:

- Trong năm 2022, tiếp tục triển khai và mở bán các dự án lớn: triển khai 3 dự án lớn là Vinhomes Ocean Park, Vinhomes Smart City và Vinhomes Grand Park (gần hoàn thành), VHM cũng đã mở bán 2 dự án mới tại Hưng Yên (một dự án giáp Hà Nội) là Vinhomes Ocean Park 2 - The Empire với tổng diện tích 445 ha và Vinhomes Ocean Park 3 – The Crown với tổng diện tích 294 ha. Hai dự án trên là các siêu đô thị với cả sản phẩm thấp tầng và cao tầng, có hạ tầng tiện ích như công viên, TTTM, bệnh viện, trường học...

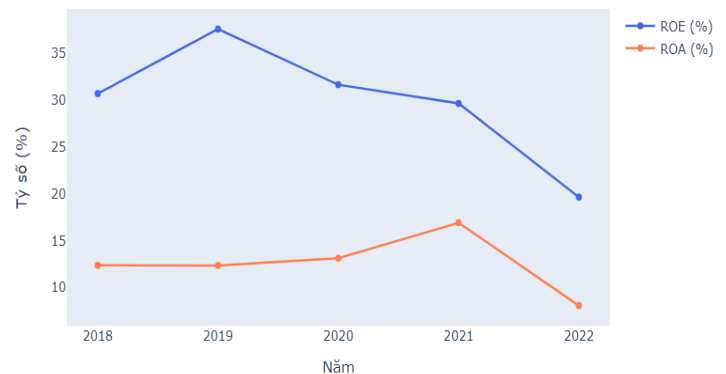
- Doanh số bán nhà trong năm 2022 đạt 128,2 nghìn tỷ đồng, tăng tới 62,5% so với năm 2021 (năm chịu ảnh hưởng của Covid-19). Kết quả đạt được chủ yếu từ dự án Vinhomes Ocean Park 2 The Empire (88,5 nghìn tỷ đồng), Vinhomes Ocean Park 3 - The Crown (20,5 nghìn tỷ đồng) và 3 dự án lớn là Ocean Park, Smart City, Grand Park với tổng giá trị 16,6 nghìn tỷ đồng.

- Dẫn đầu về doanh thu trong nhóm ngành Bất động sản, doanh thu chưa thực hiện tại thời điểm cuối năm 2022 là 107,6 nghìn tỷ đồng, tăng 105% so với năm 2021. Đối diện với sự ảnh hưởng tiêu cực của tình trạng khó khăn hiện nay của thị trường Bất động sản, VHM ít chịu ảnh hưởng hơn so với các doanh nghiệp Bất động sản khác (có doanh thu chưa thực hiện thấp và tỷ lệ vay nợ cao).

Kết quả kinh doanh theo Năm

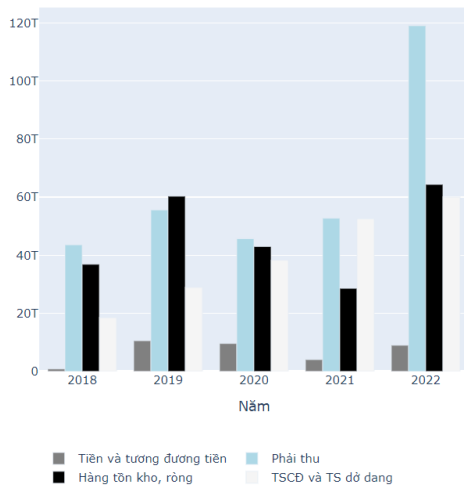


Chỉ số sinh lời

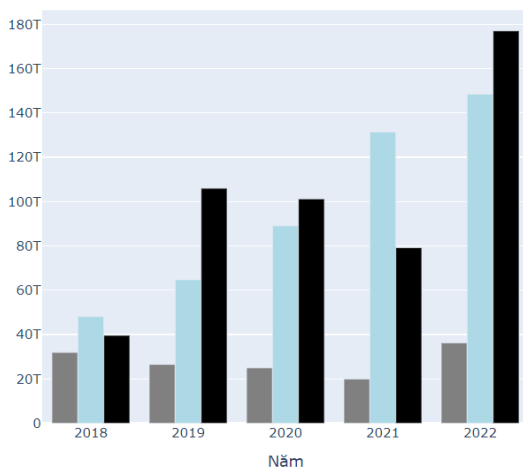


Tình hình tài chính:

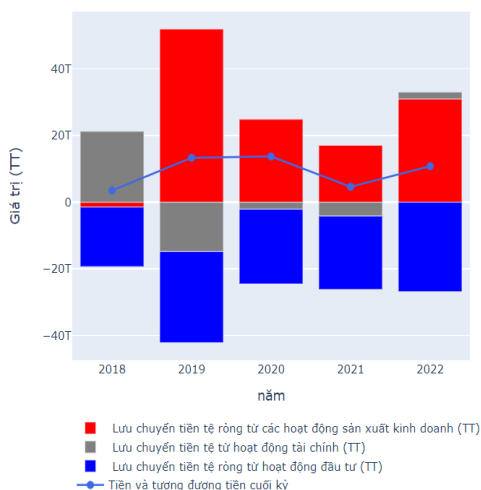
Phân bổ tài sản



Phân bổ nguồn vốn



Biểu đồ Lưu chuyển tiền tệ và Tiền tương đương cuối kỳ theo Năm



▪ **Tổng tài sản 2022 tăng** so với năm 2021, **đạt hơn 361.812 tỷ VND**. Tài sản ngắn hạn chiếm hơn 54% tổng tài sản, tài sản dài hạn chiếm gần 46%. Tài sản tập trung chủ yếu ở hàng tồn kho, quyền sử dụng đất, tiền gửi ngắn hạn, tiền và các khoản phải thu ngắn hạn.

▪ **Hàng tồn kho ròng tăng 35.738 tỷ VND**. Lượng lớn hàng tồn kho này chủ yếu bao gồm tiền sử dụng đất, chi phí giải phóng mặt bằng, chi phí xây dựng và phát triển các dự án Vinhomes Grand Park, Vinhomes Smart City, Vinhomes Ocean Park, Vinhomes Ocean Park 2 – The Empire và Vinhomes Ocean Park 3 – The Crown cũng như các chi phí và hàng hóa/ vật liệu khác. **Tỷ lệ hàng tồn kho trên tổng tài sản duy trì ở mức thấp**, chỉ chiếm 17% tổng, do đó **áp lực giải phóng hàng tồn kho của VHM không cao**.

▪ **Nợ vay chiếm 10% tổng tài sản**, trong đó nợ vay dài hạn chiếm gần 6%. **Tỷ lệ Vốn vay/ VCSH (Hệ số nợ) của VHM hiện tại khoảng gần 0.2** nghĩa chỉ có gần 20% nguồn vốn đến từ nợ vay đây là dấu hiệu của sự ổn định tài chính vì công ty không quá phụ thuộc vào nợ vay, tỷ lệ này vẫn ở mức thấp so với các công ty cùng ngành mặc dù đã tăng so với năm 2021. Các tỷ số thanh toán hiện hành, tỷ số thanh toán lãi vay duy trì ở mức ổn định cho thấy **công ty vẫn có khả năng chi trả cho các khoản nợ của mình**.

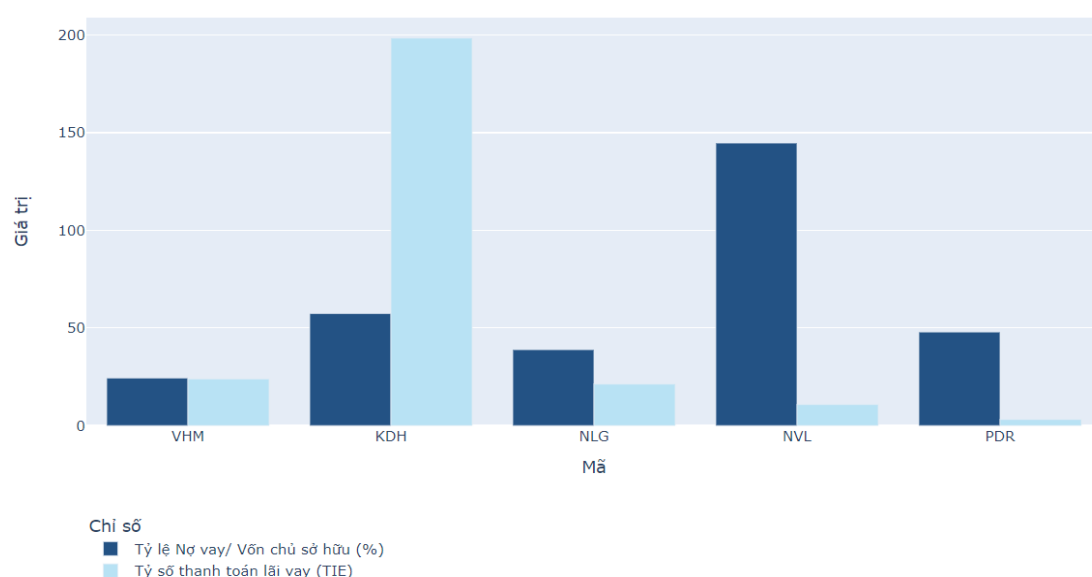
▪ **Lưu chuyển tiền ròng từ hoạt động kinh doanh năm 2022 tăng mạnh** so với năm 2021 mặc dù ngành Bất động sản đang trong thời kỳ ảm đạm, VHM

vẫn luôn duy trì dòng tiền từ hoạt động kinh doanh dương qua các năm có thể thấy sức khỏe doanh nghiệp khá bền vững.

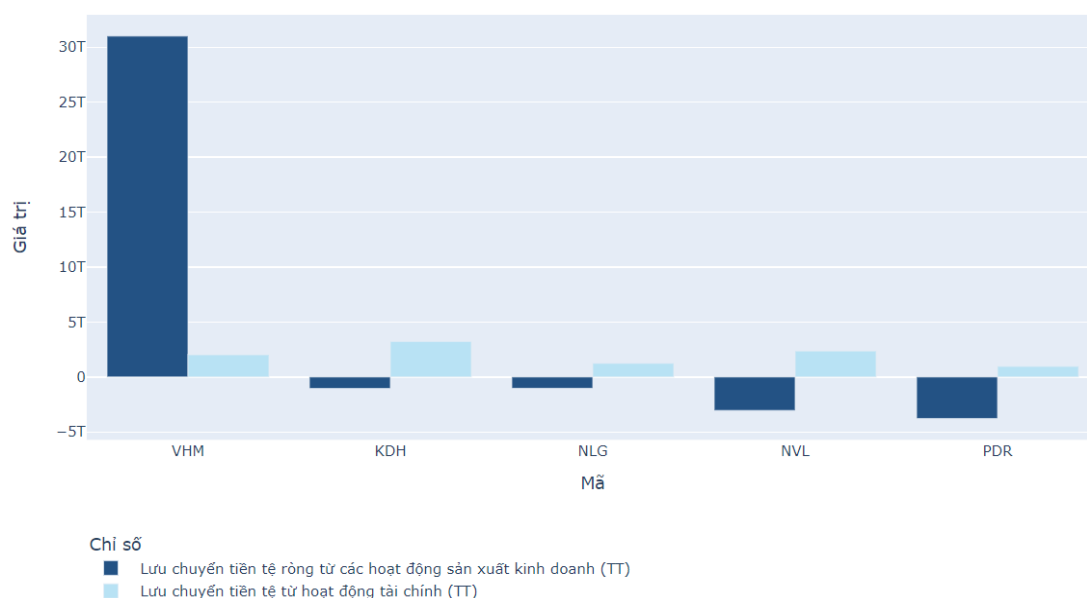
▪ **Tỷ lệ nợ vay/vốn chủ sở hữu (VCSH) của VHM vẫn ở mức thấp** so với các công ty cùng ngành. Tính đến năm 2022, tỷ lệ vốn vay/vốn chủ sở hữu (VCSH) của VHM là **24%**, đây là **mức khá an toàn** đối với một nhà phát triển Bất động sản.

▪ Trong cơ cấu nợ (tổng nợ 40,2 nghìn tỷ đồng), 24% đến từ trái phiếu trong nước, 36% đến từ vay ngân hàng, 39% còn lại đến từ vay đối tác và các nguồn khác. VHM tiếp tục tập trung vay VND, chiếm 100% tổng dư nợ, giúp **loại bỏ hoàn toàn rủi ro tỷ giá** đối với khoản vay của VHM.

So sánh Tỷ lệ Nợ vay/ Vốn chủ sở hữu (%) và TIE trong năm 2022



So sánh Dòng tiền từ hoạt động kinh doanh, tài chính năm 2022



**Một số chỉ tiêu tài chính quan trọng:*

Các chỉ tiêu tài chính trọng yếu	2019	2020	2021	2022
Báo cáo kết quả kinh doanh (tỷ đồng)				
Doanh thu	51.626	71.546	84.985	62.392
Chi phí lãi vay	2.377	3.001	2.348	1.075
Lợi nhuận trước thuế	29.745	36.516	48.183	38.642
Lợi nhuận sau thuế	24.319	28.206	38.948	29.161
Bảng cân đối kế toán (tỷ đồng)				
Tiền và tương đương tiền	1.333	1.371	4.625	10.817
Tổng nợ vay	26.505	24.955	19.919	36.206
Vốn chủ sở hữu	64.715	89.129	131.407	148.522
Tổng tài sản	197.241	215.326	230.516	361.813
Báo cáo lưu chuyển tiền tệ (tỷ đồng)				
Dòng tiền từ hoạt động kinh doanh	51.928	24.882	17.073	31.003
Chỉ số tài chính				
Lợi nhuận sau thuế/ Doanh thu (%)	47.10	38.2	45.7	46.2
Tổng nợ vay/ Vốn chủ sở hữu (x)	0.13	0.12	0.1	0.2
Tỉ số thanh toán lãi vay (x)	11.51	11.16	19.51	17.61
ROA (%)	12.9	13.3	17.4	9.7
ROE (%)	37.57	31.64	29.63	19.63

Đánh giá rủi ro:

- **Rủi ro pháp lý:** Các văn bản pháp lý còn chồng chéo, ví dụ Luật Đất đai và Luật Kinh doanh BĐS chưa đồng bộ làm chậm trễ quá trình phê duyệt dự án và ảnh hưởng hiệu quả các dự án. Tuy nhiên các quy định mới được kỳ vọng sẽ tháo gỡ khó khăn trong phê duyệt giúp thị trường sôi động trở lại.

- **Rủi ro trong kinh doanh:** Việc thuộc cơ cầu tập đoàn của Vingroup có thể khiến Vinhomes chịu một số rủi ro từ Tập đoàn. Nếu một doanh nghiệp khác thuộc Tập đoàn gặp khó khăn tài chính, việc hoàn trả tiền vay cho Vinhomes có thể bị chậm trễ hoặc không thể thực hiện, điều này có thể khiến tình hình tài chính của VHM nhanh chóng trở nên căng thẳng.

- **Rủi ro khi triển khai dự án:** Các dự án trong kế hoạch phát triển của Công ty hầu hết là các siêu dự án đòi hỏi chuyên môn đặc biệt và thường có rủi ro cao hơn trong quá trình thực hiện. Chỉ một sự kiện không mong muốn cũng có thể làm gián đoạn quá trình triển khai, dẫn đến việc động vẫn có thể diễn ra trong thời gian dài, từ đó gây ra những vấn đề về dòng tiền cho Công ty. Tuy nhiên, so với các doanh nghiệp Bất động sản khác, VHM có chuyên môn tốt hơn rất nhiều về quản trị các siêu dự án.

*File Code:

```
import dash
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input, Output
import plotly.express as px
import pandas as pd
from plotly.subplots import make_subplots
import plotly.graph_objs as go
df_price = pd.read_excel('Price-Vol VN 2015-2023.xlsx', sheet_name='Price')
**So sánh công ty cùng ngành**
def process_stock_data(df, code):
    stock_data = df[df['Code'] == code]
    stock_data['Code'] = stock_data['Code'].str.replace('VT:', '').str.replace('(P)',
    '')
    stock_data = stock_data.melt(id_vars=['Name', 'Code', 'CURRENCY'],
                                value_vars=stock_data.columns[2:-6],
                                var_name='Date',
                                value_name='Stock_Price')
    stock_data.sort_values(by=['Code', 'Date'], inplace=True)
    stock_data['Date'] = pd.to_datetime(stock_data['Date'], format='%Y-%m-%d')
    stock_data = stock_data.dropna(subset=['Stock_Price'])
    return stock_data

vhm = process_stock_data(df_price, 'VT:VHM(P)')
nvl = process_stock_data(df_price, 'VT:NVL(P)')
pdr = process_stock_data(df_price, 'VT:PDR(P)')
kdh = process_stock_data(df_price, 'VT:KDH(P)')
nlg = process_stock_data(df_price, 'VT:NLG(P)')
#%%
def price_comparison_chart():
    fig = go.Figure()

    start_date = '2018-01-01'
    vhm_filtered = vhm[vhm['Date'] >= start_date]
    nvl_filtered = nvl[nvl['Date'] >= start_date]
    kdh_filtered = kdh[kdh['Date'] >= start_date]
    pdr_filtered = pdr[pdr['Date'] >= start_date]
    nlg_filtered = nlg[nlg['Date'] >= start_date]

    fig.add_trace(go.Scatter(x=vhm_filtered['Date'], y=vhm_filtered['Stock_Price'],
mode='lines', name='VHM'))
    fig.add_trace(go.Scatter(x=nvl_filtered['Date'], y=nvl_filtered['Stock_Price'],
mode='lines', name='NVL'))
    fig.add_trace(go.Scatter(x=kdh_filtered['Date'], y=kdh_filtered['Stock_Price'],
mode='lines', name='KDH'))
    fig.add_trace(go.Scatter(x=pdr_filtered['Date'], y=pdr_filtered['Stock_Price'],
mode='lines', name='PDR'))
    fig.add_trace(go.Scatter(x=nlg_filtered['Date'], y=nlg_filtered['Stock_Price'],
mode='lines', name='NLG'))

    fig.update_layout(
        xaxis_title='Ngày',
        yaxis_title='Giá cổ phiếu',
        title='Biểu đồ giá của VHM, KDH, NVL, PDR và NLG'
    )
    return dcc.Graph(figure=fig)
#%%
def process_excel_file(file_path, year):
    df = pd.read_excel(file_path, skiprows=8, header=0)
```

```

cols_to_drop = [
    'Sàn', 'Ngành ICB - cấp 1', 'Ngành ICB - cấp 2', 'Ngành ICB - cấp 3', 'STT',
    f'Quý\nHợp nhất\nQuý: Hàng năm\nNăm: {year}\n',
    f'Năm\nHợp nhất\nQuý: Hàng năm\nNăm: {year}\n',
    f'Trạng thái kiểm toán\nHợp nhất\nQuý: Hàng năm\nNăm: {year}\n'
]
df = df.drop(columns=cols_to_drop)

df.columns = df.columns.str.replace("CBKT.", "").str.replace("TM.",
"").str.replace(
    f"\nHợp nhất\nQuý: Hàng năm\nNăm: {year}\nĐơn vị: Triệu VND",
"").str.replace("KQKD.", "").str.replace("LCTT.", "")
return df

df_2022 = process_excel_file("2022-Vietnam.xlsx", 2022)
df_2018 = process_excel_file("2021-Vietnam.xlsx", 2018)
df_2019 = process_excel_file("2020-Vietnam.xlsx", 2019)
df_2021 = process_excel_file("2018-Vietnam.xlsx", 2021)
df_2020 = process_excel_file("2019-Vietnam.xlsx", 2020)
###
df_comparison_2022 = df_2022[df_2022['Ngành ICB - cấp 4'] == 'Bất động sản']
df_comparison_2022['Tỷ lệ Nợ/Tổng tài sản (%)'] = (df_comparison_2022['NỢ PHẢI TRẢ']
/ df_comparison_2022['TỔNG CỘNG TÀI SẢN']) * 100
df_comparison_2022['EBIT'] = df_comparison_2022['Lợi nhuận gộp về bán hàng và cung
cấp dịch vụ'] - df_comparison_2022['Chi phí bán hàng'] - df_comparison_2022['Chi
phí quản lý doanh nghiệp'] + df_comparison_2022['Doanh thu hoạt động tài chính
khác']
df_comparison_2022['Tỷ số thanh toán lãi vay (TIE)'] = df_comparison_2022['EBIT'] /
( - df_comparison_2022['Trong đó: Chi phí lãi vay'])
df_comparison_2022['ROE (%)'] = (df_comparison_2022['Lợi nhuận sau thuế thu nhập
doanh nghiệp'] / df_comparison_2022['VỐN CHỦ SỞ HỮU']) * 100

selected_companies = ['VHM', 'KDH', 'NVL', 'PDR', 'NLG']
df_selected = df_comparison_2022[df_comparison_2022['Mã'].isin(selected_companies)]
###
def debt_chart():
    fig = go.Figure()
    fig = px.bar(df_selected, x='Mã', y=['Tỷ lệ Nợ/Tổng tài sản (%)', 'Tỷ số thanh
toán lãi vay (TIE)'],
                title='So sánh Tỷ lệ Nợ/ Tổng tài sản và TIE trong năm 2022',
                labels={'variable': 'Chi số', 'value': 'Giá trị'},
                color_discrete_sequence=['#235284', '#b8e2f4', '#000000'],
                barmode='group')

    fig.show()
    return dcc.Graph(figure=fig)
###
def cash_flow():
    fig = go.Figure()
    fig = px.bar(df_selected, x='Mã', y=['Lưu chuyển tiền tệ ròng từ các hoạt động
sản xuất kinh doanh (TT)',
                                         'Lưu chuyển tiền tệ từ hoạt động tài chính
(TT)'],
                title='So sánh Dòng tiền từ hoạt động kinh doanh, tài chính năm
2022',
                labels={'variable': 'Chi số', 'value': 'Giá trị'},
                color_discrete_sequence=['#235284', '#b8e2f4'],
                barmode='group')

    fig.show()
    return dcc.Graph(figure=fig)
###

```

```

def revenue_comparison():
    df_comparison_2022['Tỷ trọng doanh thu'] = round(df_comparison_2022[' Lợi nhuận
sau thuế thu nhập doanh nghiệp'] /
                                                    df_comparison_2022[' Lợi nhuận
sau thuế thu nhập doanh nghiệp'].sum() * 100, 1)
    fig = go.Figure()
    df_2022_sorted = df_comparison_2022.sort_values(by='Tỷ trọng doanh thu',
ascending=False)
    fig = px.pie(df_2022_sorted, names='Mã', values='Tỷ trọng doanh thu',
                title='Tỷ trọng doanh thu của các công ty trong tổng ngành')
    top_5_labels = df_2022_sorted.head(5)['Mã']
    df_2022_sorted['labels'] = df_2022_sorted.apply(lambda row: f"{row['Mã']}
{row['Tỷ trọng doanh thu']}"
                                                    if row['Mã'] in top_5_labels.values else '',
axis=1)
    fig.update_traces(textinfo='label', textposition='inside',
                      insidetextorientation='radial',
                      text=df_2022_sorted['labels'])
    fig.update_layout(
        title_text='Tỷ trọng doanh thu của các công ty trong ngành bất động sản
(2022)',
        showlegend=False
    )
    return dcc.Graph(figure=fig)
#%%
def generate_industry_comparison():
    app = dash.Dash(__name__)

    app.layout = html.Div([
        html.H1("So sánh các công ty dẫn đầu trong ngành Bất động sản"),
        html.Div([
            html.Div([
                price_comparison_chart()
            ], className='six columns'),
            html.Div([
                revenue_comparison()
            ], className='six columns')
        ], className='row'),
        html.Div([
            html.Div([
                cash_flow()
            ], className='four columns'),
        ], className='row'),
        html.Div([
            html.Div([
                debt_chart()
            ], className='four columns')
        ], className='row')
    ])

    return app.layout
**Phân tích báo cáo tài chính**
def select_vhm(df, year):
    selected_df = df[df['Mã'] == 'VHM']
    selected_df.insert(0, 'Năm', year)
    return selected_df

df_vhm_2022 = select_vhm(df_2022, 2022)
df_vhm_2020 = select_vhm(df_2020, 2020)
df_vhm_2019 = select_vhm(df_2019, 2019)
df_vhm_2018 = select_vhm(df_2018, 2018)

```

```

df_vhm_2021 = select_vhm(df_2021, 2021)

df_vhm = pd.concat([df_vhm_2021, df_vhm_2020, df_vhm_2019, df_vhm_2018, df_vhm_2022])
df_vhm = df_vhm.sort_values(by='Năm')
#%%
def asset():
    fig = go.Figure()

    fig.add_trace(go.Bar(x=df_vhm['Năm'], y=df_vhm['Tiền và tương đương tiền'],
name='Tiền và tương đương tiền',
marker_color='#808080'))
    fig.add_trace(go.Bar(x=df_vhm['Năm'], y=df_vhm['Các khoản phải thu ngắn hạn'] +
df_vhm['Phải thu dài hạn'], name='Phải thu',
marker_color='#add8e6'))
    fig.add_trace(go.Bar(x=df_vhm['Năm'], y=df_vhm['Hàng tồn kho, ròng'], name='Hàng
tồn kho, ròng',
marker_color='#000000'))
    fig.add_trace(go.Bar(x=df_vhm['Năm'], y=df_vhm['Tài sản cố định'] +df_vhm['Tài
sản dở dang dài hạn'], name='TSCĐ và TS dở dang',
marker_color='#F5F5F5'))

    fig.update_layout(
        title='Phân bổ tài sản',
        xaxis=dict(title='Năm', tickmode='array', tickvals= df_vhm['Năm']))
    return dcc.Graph(figure = fig)
#%%
def cash():
    columns_to_plot = [
        'Luu chuyển tiền tệ ròng từ các hoạt động sản xuất kinh doanh (TT)',
        'Luu chuyển tiền tệ từ hoạt động tài chính (TT)',
        'Luu chuyển tiền tệ ròng từ hoạt động đầu tư (TT)',
        'Tiền và tương đương tiền cuối kỳ (TT)']
    colors = {'Luu chuyển tiền tệ ròng từ các hoạt động sản xuất kinh doanh (TT)':
'#FF0000',
        'Luu chuyển tiền tệ từ hoạt động tài chính (TT)': '#808080',
        'Luu chuyển tiền tệ ròng từ hoạt động đầu tư (TT)': '#0000FF'}
    fig = go.Figure()
    for column in columns_to_plot[:-1]:
        fig.add_trace(go.Bar(x=df_vhm['Năm'], y=df_vhm[column],
name=column,marker_color=colors[column]))

    fig.add_trace(go.Scatter(
        x=df_vhm['Năm'], y=df_vhm['Tiền và tương đương tiền cuối kỳ (TT)'],
mode='lines+markers', name='Tiền và tương đương tiền cuối kỳ',
line=dict(color='royalblue', width=2),
marker=dict(color='royalblue', size=8)
    ))

    fig.update_layout(
        barmode='relative', title='Biểu đồ Lưu chuyển tiền tệ và Tiền tương đương
cuối kỳ theo Năm',
        xaxis_title='năm', yaxis_title='Giá trị (TT)',)

    fig.update_traces(hoverinfo="name+y", hovertemplate=None)
    return dcc.Graph(figure =fig)
#%%
def profit_growth():
    df_vhm['Biên lợi nhuận ròng'] = (df_vhm['Lợi nhuận sau thuế thu nhập doanh
nghiệp']/ df_vhm['Doanh thu thuần'])*100
    fig = go.Figure()
    fig.add_trace(go.Bar(
        x=df_vhm['Năm'],

```

```

        y=df_vhm[' Doanh thu thuần'],
        name='Doanh thu',
        marker_color='#808080',
    ))
fig.add_trace(go.Bar(
    x=df_vhm['Năm'],
    y=df_vhm[' Lợi nhuận sau thuế thu nhập doanh nghiệp'],
    name='Lợi nhuận sau thuế',
    marker_color='#0000FF',
))
fig.add_trace(go.Scatter(
    x=df_vhm['Năm'],
    y=df_vhm['Biên lợi nhuận ròng'],
    name='Biên lợi nhuận ròng',
    mode='lines+markers',
    line=dict(color='#FF0000'),
    yaxis='y2',
))
fig.update_layout(
    title='Kết quả kinh doanh theo Năm',
    xaxis=dict(title='Năm', tickmode='array', tickvals= df_vhm['Năm']),
    yaxis=dict(title=''),
    yaxis2=dict(title='', overlaying='y', side='right', showgrid=False, range=[0,
50]),
    barmode='group')
return dcc.Graph(figure=fig)
"""
def equity():
    fig = go.Figure()
    fig.add_trace(go.Bar(x=df_vhm['Năm'], y=df_vhm[' Vay và nợ thuê tài chính ngắn
hạn'] +df_vhm[' Vay và nợ thuê tài chính dài hạn'], name='Vay nợ',
        marker_color='#808080'))
    fig.add_trace(go.Bar(x=df_vhm['Năm'], y=df_vhm[' VỐN CHỦ SỞ HỮU'], name='Vốn
CSH',
        marker_color='#add8e6'))
    fig.add_trace(go.Bar(x=df_vhm['Năm'], y=df_vhm[' NỢ PHẢI TRẢ'] - (df_vhm[' Vay và
nợ thuê tài chính ngắn hạn'] +df_vhm[' Vay và nợ thuê tài chính dài hạn']), name='Nợ
khác',
        marker_color='#000000'))
    fig.update_layout(title='Phân bổ nguồn vốn',
        xaxis=dict(title='Năm', tickmode='array', tickvals=
df_vhm['Năm'])
    )
    return dcc.Graph(figure = fig)
"""
def debt_asset_ratio():
    df_vhm['Hệ số nợ'] = (df_vhm[' Vay và nợ thuê tài chính ngắn hạn'] +df_vhm[' Vay
và nợ thuê tài chính dài hạn']) / df_vhm[' TỔNG CỘNG NGUỒN VỐN']
    df_vhm['EBIT'] = df_vhm[' Tổng lợi nhuận kế toán trước thuế'] + df_vhm[' Trong
đó: Chi phí lãi vay']
    df_vhm['Tỷ số thanh toán lãi vay (TIE)'] = df_vhm['EBIT'] / ( - df_vhm[' Trong
đó: Chi phí lãi vay'])
    df_vhm['Hệ số khả năng thanh toán hiện hành'] = df_vhm[' TÀI SẢN NGẮN HẠN']/
df_vhm[' Nợ ngắn hạn']
    fig = go.Figure()
    fig.add_trace(go.Scatter(
        x=df_vhm['Năm'],
        y=df_vhm['Hệ số nợ'],
        mode='lines+markers',
        name='Hệ số nợ',
        line=dict(color='#00429d'),

```

```

))
fig.add_trace(go.Scatter(
    x=df_vhm['Năm'],
    y=df_vhm['Tỷ số thanh toán lãi vay (TIE)'],
    mode='lines+markers',
    name='Tỷ số thanh toán lãi vay (TIE)',
    line=dict(color='coral'),
))
fig.add_trace(go.Scatter(
    x=df_vhm['Năm'],
    y=df_vhm['Hệ số khả năng thanh toán hiện hành'],
    mode='lines+markers',
    name='Hệ số khả năng thanh toán hiện hành',
    line=dict(color='#d64161'),
))
fig.update_layout(
    title='Sức khỏe tài chính',
    xaxis=dict(title='Năm', tickmode='array', tickvals=df_vhm['Năm'].unique(),
showgrid=False),
    yaxis=dict(showgrid=False),
)
return dcc.Graph(figure=fig)
#%%
def generate_financial_analysis():
    app = dash.Dash(__name__)

    app.layout = html.Div([
        html.H1("Biểu đồ phân tích tài chính"),
        html.Div([
            html.Div([
                asset()
            ], className='six columns'),
            html.Div([
                equity()
            ], className='six columns')
        ], className='row'),
        html.Div([
            cash()
        ], className='six columns'),
        html.Div([
            html.Div([
                profit_growth()
            ], className='four columns'),
            html.Div([
                debt_asset_ratio()
            ], className='four columns')
        ], className='row')
    ])

    return app.layout
**Phân tích kỹ thuật **
vhm_price = df_price[df_price['Code'] == 'VT:VHM(P)']
vhm_price['Code'] = vhm_price['Code'].str.replace('VT:', '').str.replace('(P)', '')
df = vhm_price.melt(id_vars=['Name', 'Code', 'CURRENCY'],
                    value_vars=vhm_price.columns[2:-6],
                    var_name='Date',
                    value_name='Stock_Price')
df.sort_values(by=['Code', 'Date'], inplace=True)
df['Date'] = pd.to_datetime(df['Date'], format='%Y-%m-%d')
df = df.dropna(subset=['Stock_Price'])
#%%

```

```

def price_vhm():
    fig = go.Figure()
    fig.add_trace(go.Scatter(x=df['Date'], y=df['Stock_Price'], mode='lines',
name='Giá đóng cửa'))
    fig.update_layout(
        xaxis_title='Date',
        yaxis_title='Giá đóng cửa',
        title='Biểu đồ giá của VHM',
    )
    return dcc.Graph(figure=fig)
#%%
def generate_rsi_chart():
    fig = go.Figure()
    fig = make_subplots(rows=2, cols=1, shared_xaxes=True,
vertical_spacing=0.1)

    trace1 = go.Scatter(x=df['Date'], y=df['Stock_Price'], mode='lines',
name='Giá Đóng Cửa')
    fig.add_trace(trace1, row=1, col=1)
    df['Date'] = pd.to_datetime(df['Date'])
    # Tính RSI
    delta = df['Stock_Price'].diff(1)
    gain = (delta.where(delta > 0)).fillna(0)
    loss = (-delta.where(delta < 0)).fillna(0)
    avg_gain = gain.rolling(window=14).mean()
    avg_loss = loss.rolling(window=14).mean()
    rs = avg_gain / avg_loss
    rsi = 100 - (100 / (1 + rs))

    trace2 = go.Scatter(x=df['Date'], y=rsi, mode='lines', name='RSI',
line=dict(width=1))
    fig.add_trace(trace2, row=2, col=1)

    fig.update_xaxes(type='date', tickmode='auto', row=2, col=1)
    fig.update_yaxes(title_text="RSI", row=2, col=1, range=[0, 100])

    fig.update_xaxes(type='date', tickmode='auto', row=1,
col=1, showticklabels=False )
    fig.update_yaxes(title_text="Giá", row=1, col=1)
    fig.update_layout(width=1000, height=600)
    fig.update_layout(title=f"Biểu đồ giá và RSI của VHM")
    return dcc.Graph(figure=fig)
#%%
def generate_macd_chart():
    fig = make_subplots(rows=2, cols=1, shared_xaxes=True, vertical_spacing=0.1)
    df['Date'] = pd.to_datetime(df['Date'])
    trace1 = go.Scatter(x=df['Date'], y=df['Stock_Price'], mode='lines', name='Giá
Đóng Cửa')
    fig.add_trace(trace1, row=1, col=1)

    short_ema = df['Stock_Price'].ewm(span=12, adjust=False).mean()
    long_ema = df['Stock_Price'].ewm(span=26, adjust=False).mean()
    macd = short_ema - long_ema
    signal = macd.ewm(span=9, adjust=False).mean()
    histogram = macd - signal

    trace2 = go.Scatter(x=df['Date'], y=macd, mode='lines', name='MACD')
    trace3 = go.Scatter(x=df['Date'], y=signal, mode='lines', name='Signal Line')
    trace4 = go.Bar(x=df['Date'], y=histogram, name='Histogram', base=0)

    fig.add_trace(trace2, row=2, col=1)

```



```

fig.add_trace(trace3, row=2, col=1)
fig.add_trace(trace4, row=2, col=1)

fig.update_xaxes(type='category', tickvals=df['Date'].dt.strftime('%Y-%m-%d'), row=2,
col=1, showticklabels=True)
fig.update_yaxes(title_text="Giá trị MACD", row=2, col=1)

fig.update_xaxes(type='category', tickvals=df['Date'].dt.strftime('%Y-%m-%d'), row=1,
col=1, showticklabels=False)
fig.update_yaxes(title_text="Giá", row=1, col=1)
fig.update_yaxes(range=[macd.min(), macd.max()], row=2, col=1, secondary_y=True)
fig.update_layout(title=f"Biểu đồ giá và MACD của VHM")
return dcc.Graph.figure=fig)

#%%
def generate_sma_chart():
    global df
    fig= go.Figure()
    df['SMA30'] = df['Stock_Price'].rolling(window=30).mean()
    df = df.dropna(subset=['SMA30'])

    fig.add_trace(go.Scatter(x=df['Date'], y=df['Stock_Price'], mode='lines',
name='Stock Price'))
    fig.add_trace(go.Scatter(x=df['Date'], y=df['SMA30'], mode='lines',
name='SMA30'))

    fig.update_xaxes({'showline': False, 'showgrid': False})
    fig.update_yaxes({'showline': False, 'showgrid': False, 'title': 'SMA30'})
    # Cấu hình tiêu đề và hiển thị biểu đồ
    fig.update_layout(title=f"Biểu đồ giá và SMA30 của VHM")
    return dcc.Graph.figure=fig)

#%%
def generate_bollinger_chart():
    global df
    fig = go.Figure()
    df['SMA'] = df['Stock_Price'].rolling(window=20).mean()
    df['std'] = df['Stock_Price'].rolling(window=20).std()
    df['Upper'] = df['SMA'] + 2 * df['std']
    df['Lower'] = df['SMA'] - 2 * df['std']
    df = df.dropna(subset=['Upper'])
    df = df.dropna(subset=['Lower'])
    df = df.dropna(subset=['SMA'])
    fig.add_trace(go.Scatter(x=df['Date'], y=df['Stock_Price'], mode='lines',
name='Stock Price'))
    fig.add_trace(go.Scatter(x=df['Date'], y=df['Upper'], mode='lines', name='Upper
Bollinger Band',
                                line=dict(color='#1f77b4'))))
    fig.add_trace(go.Scatter(x=df['Date'], y=df['SMA'], mode='lines', name='Middle
Bollinger Band',
                                line=dict(color='orange'))))
    fig.add_trace(go.Scatter(x=df['Date'], y=df['Lower'], mode='lines', name='Lower
Bollinger Band',
                                line=dict(color='#1f77b4'))))

    fig.update_layout(
        xaxis={'showline': False, 'showgrid': False},
        yaxis={'showline': False, 'showgrid': False},
        yaxis2={'showline': False, 'showgrid': False}
    )
    fig.update_traces(marker=dict(line=dict(width=6)))
    fig.update_layout(width=1000, height=600)

```

```

fig.update_layout(barmode='overlay', bargap=1)
fig.update_layout(title=f"Biểu đồ giá và Bollinger Bands của VHM")
return dcc.Graph(figure=fig)
**Tổng quan công ty**
def generate_company_overview():
    return html.Div([
        html.Details([
            html.Summary("Hồ sơ công ty", style={'fontWeight': 'bold', 'fontSize':
'large'})),
            html.Div([
                html.P("Tên đầy đủ: CTCP Vinhomes"),
                html.P("Tên tiếng Anh: Vinhomes JSC"),
                html.P("Tên viết tắt: Vinhomes., JSC"),
                html.P("CTCP Vinhomes có tiền thân là Công ty Cổ phần Đô thị BIDV-PP
được thành lập năm 2008 với vốn điều lệ ban đầu là 300 tỷ đồng. Đây là thương hiệu
bất động sản trực thuộc Tập đoàn VinGroup. Năm 2018, cổ phiếu VHM giao dịch trên Sở
Giao dịch Chứng khoán Thành phố Hồ Chí Minh (HOSE). Công ty hoạt động chính trong các
lĩnh vực Đầu tư, phát triển và kinh doanh bất động sản nhà ở, văn phòng và các hạng
mục liên quan; Chuẩn bị mặt bằng, lắp đặt trang thiết bị và hoàn thiện công trình xây
dựng."),
            ])
        ]),
        html.Details([
            html.Summary("Niêm yết", style={'fontWeight': 'bold', 'fontSize':
'large'})),
            html.Div([
                html.P("Sàn giao dịch: HOSE"),
                html.P("Nhóm ngành: Xây dựng và Bất động sản"),
                html.P("Ngành: Phát triển bất động sản"),
                html.P("Ngày niêm yết: 17/05/2018")
            ])
        ]),
        html.Details([
            html.Summary("Mốc lịch sử", style={'fontWeight': 'bold', 'fontSize':
'large'})),
            html.Div([
                html.P("- Năm 2008: Tiền thân Công ty Cổ phần Đô thị BIDV-PP được
thành lập với vốn điều lệ ban đầu là 300 tỷ đồng."),
                html.P("- Tháng 03/2009: Đổi tên thành Công ty Cổ phần Phát triển Đô
thị Nam Hà Nội."),
                html.P("- Ngày 26/06/2009: Tăng vốn điều lệ lên 500 tỷ đồng."),
                html.P("- Ngày 05/08/2010: Tăng vốn điều lệ lên 2,000 tỷ đồng."),
                html.P("- Ngày 28/10/2011: Trở thành công ty đại chúng theo Công văn
số 3585/UBCK-QLPH của Ủy ban chứng khoán Nhà nước."),
                html.P("- Ngày 03/11/2011: Công ty nhận được Quyết định về việc chấp
thuận đăng ký giao dịch cổ phiếu tại Sở Giao dịch Chứng khoán Hà Nội (UPCoM) với mã
chứng khoán là NHN."),
                html.P("- Ngày 14/07/2017: Hủy đăng ký giao dịch trên thị trường
UPCoM."),
                html.P("- Ngày 01/02/2018: Tăng vốn điều lệ lên 28,365 tỷ đồng."),
                html.P("- Ngày 02/02/2018: Đổi tên thành Công ty Cổ phần Vinhomes."),
                html.P("- Ngày 14/07/2017: Hủy đăng ký giao dịch trên thị trường
UPCoM."),
                html.P("- Ngày 12/02/2018: Giảm vốn điều lệ xuống 26,796 tỷ đồng."),
                html.P("- Ngày 07/05/2018: Niêm yết trên Sở Giao dịch Chứng khoán
Thành phố Hồ Chí Minh (HOSE)."),
                html.P("- Ngày 17/05/2018: là ngày giao dịch đầu tiên trên sàn HOSE
với giá tham chiếu là 92,100 đ/CP."),
                html.P("- Ngày 08/11/2018: Tăng vốn điều lệ lên 33,495 tỷ đồng."),
                html.P("- Tháng 10/2021: Tăng vốn điều lệ lên 43,543 tỷ đồng."),
            ])
        ])
    ])

```

```

    ]),
    html.Details([
        html.Summary("Liên hệ", style={'fontWeight': 'bold', 'fontSize':
'large'})),
        html.Div([
            html.P("Địa chỉ: Tòa nhà văn phòng Symphony - Đường Chu Huy Mân - KĐT
Sinh thái Vinhomes Riverside - P. Phúc Lợi - Q. Long Biên - Tp. Hà Nội"),
            html.P("Điện thoại: (84.24) 3974 9350"),
            html.P("Fax: (84.24) 3974 9351"),
            html.P("Email: info@vinhomes.vn"),
            html.P("Website: https://vinhomes.vn/"),
        ])
    ]),
])
**Tạo web Dash**
app = dash.Dash(__name__)

app.layout = html.Div([
    html.H1("Phân tích công ty Vinhomes (Mã: VHM)"),
    dcc.Tabs(id="tabs", value='tab-1', children=[
        dcc.Tab(label='Tổng quan', value='tab-1'),
        dcc.Tab(label='Phân tích kỹ thuật', value='tab-2'),
        dcc.Tab(label='Tài chính', value='tab-3'),
        dcc.Tab(label='DN cùng ngành', value='tab-4'),
    ]),
    html.Div(id='tabs-content'),
    html.Div([
        dcc.Dropdown(
            id='indicator-dropdown',
            options=[
                {'label': 'Price', 'value': 'price'},
                {'label': 'SMA30', 'value': 'sma30'},
                {'label': 'Bollinger Bands', 'value': 'bollinger'},
                {'label': 'MACD', 'value': 'macd'},
                {'label': 'RSI', 'value': 'rsi'},
            ],
            placeholder="Chọn một chỉ số",
            style={'display': 'none'}
        )
    ], id='dropdown-container'),
    html.Div(id='technical-analysis-content', style={'display': 'none'})
])

@app.callback(
    Output('tabs-content', 'children'),
    [Input('tabs', 'value')]
)
def render_tab_content(tab):
    if tab == 'tab-1':
        return generate_company_overview()
    elif tab == 'tab-2':
        return html.Div([
            dcc.Dropdown(
                id='indicator-dropdown',
                options=[
                    {'label': 'Price', 'value': 'price'},
                    {'label': 'SMA30', 'value': 'sma30'},
                    {'label': 'Bollinger Bands', 'value': 'bollinger'},
                    {'label': 'MACD', 'value': 'macd'},
                    {'label': 'RSI', 'value': 'rsi'},
                ]
            ),

```

```

        placeholder="Chọn một chỉ số"
    ),
    html.Div(id='technical-analysis-content')
])
elif tab == 'tab-3':
    return generate_financial_analysis()
elif tab == 'tab-4':
    return generate_industry_comparison()
else:
    return html.Div("This tab has no content")

#%%
@app.callback(
    Output('technical-analysis-content', 'children'),
    [Input('indicator-dropdown', 'value')],
    prevent_initial_call=True
)
def update_technical_analysis(selected_indicator):
    if selected_indicator == 'macd':
        return generate_macd_chart()
    elif selected_indicator == 'rsi':
        return generate_rsi_chart()
    elif selected_indicator == 'sma30':
        return generate_sma_chart()
    elif selected_indicator == 'bollinger':
        return generate_bollinger_chart()
    elif selected_indicator == 'price':
        return price_vhm()
    else:
        return html.Div("Chọn một chỉ số")

#%%
# Run the app
if __name__ == '__main__':
    app.run_server(debug=True, port=8051)

```