In [2]:
```
pip install pandas
```

```
Requirement already satisfied: pandas in c:\users\hp\anaconda3\lib\site-packa
ges (2.0.1)Note: you may need to restart the kernel to use updated packages.

Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\hp\anaconda
3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: tzdata>=2022.1 in c:\users\hp\anaconda3\lib\si
te-packages (from pandas) (2023.3)
Requirement already satisfied: pytz>=2020.1 in c:\users\hp\anaconda3\lib\site
-packages (from pandas) (2021.3)
Requirement already satisfied: numpy>=1.20.3 in c:\users\hp\anaconda3\lib\sit
e-packages (from pandas) (1.24.3)
Requirement already satisfied: six>=1.5 in c:\users\hp\anaconda3\lib\site-pac
kages (from python-dateutil>=2.8.2->pandas) (1.16.0)
```

In [61]:

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the CSV file into a DataFrame
file_path = r"C:\Users\hp\Desktop\Yoshop\TASK 3\orders_2016-2020_Dataset (1).c
df = pd.read_csv(file_path)
df.fillna('0', inplace=True)
df['Order Date and Time Stamp'] = pd.to_datetime(df['Order Date and Time Stamp

# Function to perform EDA for "Shipping Address Differs from Billing Address"
def eda_shipping_billing_differs():
    df['Shipping_Differs_From_Billing'] = df['Shipping Street Address'] != df[
    differs_percentage = df['Shipping_Differs_From_Billing'].value_counts(norm
    labels = ['Shipping and Billing Match', 'Shipping and Billing Differ']
    sizes = [differs_percentage[False], differs_percentage[True]]
    colors = ['lightskyblue', 'lightcoral']
    plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', shadow=Tru
    plt.axis('equal')
    plt.title("Percentage of Shipping Address Differing from Billing Address")
    plt.show()

# Function to perform EDA for "Multiple Orders of the Same Item on a Chosen Da
def eda_multiple_orders_on_date():
    chosen_date = input("Enter the date for analysis (e.g., '2020-10-17'): ")
    chosen_date = pd.to_datetime(chosen_date).date()
    orders_on_chosen_date = df[df['Order Date and Time Stamp'].dt.date == chos
    multiple_orders = orders_on_chosen_date['LineItem Name'].value_counts()
    plt.figure(figsize=(10, 6))
    plt.bar(multiple_orders.index, multiple_orders.values)
    plt.xticks(rotation=90)
    plt.xlabel('Item Name')
    plt.ylabel('Number of Orders')
    plt.title('Multiple Orders of the Same Item on {}'.format(chosen_date))
    plt.show()
    max_orders = multiple_orders.max()
    print("Maximum number of orders on {}: {}".format(chosen_date, max_orders)

# Function to perform EDA for "Date with the Maximum Number of Orders"
def eda_date_with_max_orders():
    daily_order_counts = df['Order Date and Time Stamp'].dt.date.value_counts(
    max_orders_date = daily_order_counts.idxmax()
    plt.figure(figsize=(12, 6))
    plt.bar(daily_order_counts.index, daily_order_counts.values)
    plt.xticks(rotation=90)
    plt.xlabel('Date')
    plt.ylabel('Number of Orders')
    plt.title('Number of Orders for Each Date')
    plt.show()
    print("Date with the Unusually large orders: {}".format(max_orders_date))

# Function to perform EDA for "Multiple Orders to the Same Address with Differ
def eda_multiple_orders_diff_payment():
    duplicate_addresses = df[df.duplicated(subset=['Shipping Street Address'],
    multiple_orders_with_diff_payment = duplicate_addresses.groupby(['Shipping
    multiple_orders_with_diff_payment = multiple_orders_with_diff_payment[mult
    plt.figure(figsize=(12, 8))
```

```
    plt.bar(multiple_orders_with_diff_payment['Shipping Street Address'], mult
    plt.xticks(rotation=90)
    plt.xlabel('Shipping Street Address')
    plt.ylabel('Payment Method')
    plt.title('Addresses with Multiple Orders and Different Payment Methods')
    plt.show()

# Function to perform EDA for "Distribution of International Orders"
def eda_international_orders():
    international_orders = df[df['Fulfillment Status'] == 'International']
    num_international_orders = len(international_orders)
    num_non_international_orders = len(df) - num_international_orders
    labels = ['International Orders', 'Non-International Orders']
    sizes = [num_international_orders, num_non_international_orders]
    colors = ['lightskyblue', 'lightcoral']
    plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', shadow=Tru
    plt.axis('equal')
    plt.title('Distribution of International Orders')
    plt.show()

# Ask the user to enter a number from 1 to 5
user_choice = input("Enter a number from 1 to 5:"
                    "\n1. EDA for 'Shipping Address Differs from Billing Addre
                    "\n2. EDA for 'Multiple Orders of the Same Item on a Chose
                    "\n3. EDA for 'Date with the Maximum Number of Orders'"
                    "\n4. EDA for 'Multiple Orders to the Same Address with Di
                    "\n5. EDA for 'Distribution of International Orders'"
                    "\nYour choice: ")

# Perform the corresponding EDA based on user's choice
if user_choice == '1':
    print("1. EDA for 'Shipping Address Differs from Billing Address'")
    eda_shipping_billing_differs()
elif user_choice == '2':
    print("2. EDA for 'Multiple Orders of the Same Item '")
    eda_multiple_orders_on_date()
elif user_choice == '3':
    print("3. EDA for 'Date with the Maximum Number of Orders'")
    eda_date_with_max_orders()
elif user_choice == '4':
    print("4. EDA for 'Multiple Orders to the Same Address with Different Paym
    eda_multiple_orders_diff_payment()
elif user_choice == '5':
    print("5. EDA for 'Distribution of International Orders'")
    eda_international_orders()
else:
    print("Invalid choice. Please enter a number from 1 to 5.")
```

```
Enter a number from 1 to 5:
1. EDA for 'Shipping Address Differs from Billing Address'
2. EDA for 'Multiple Orders of the Same Item on a Chosen Date'
3. EDA for 'Date with the Maximum Number of Orders'
4. EDA for 'Multiple Orders to the Same Address with Different Payment Metho
d'
5. EDA for 'Distribution of International Orders'
Your choice: 1
1. EDA for 'Shipping Address Differs from Billing Address'
```