

# Analytical Case Study

**“A Comprehensive Analysis of Sales Data for Strategic Decision-Making”**

Presented by: Hayah Ahmed



**Introduction**

**Data Exploration**

**Analytical Questions**

**Visualisations**

**Additional  
Insights/findings**

**Recommendations**



**Important:**

- I've written SQL queries using the Oracle Application Express (APEX) development environment.
- Visualizations for the retrieved data are created using Python libraries.
- To keep things organized, the code for these functionalities is stored in separate files.
- else refer to this repo:

<https://github.com/hayahahmed08/BAZAAR-SUBMISSION>

# Agenda



# Case Study Overview

In this case study, I analyzed sales transactions from a retail company to improve business performance. I employed SQL querying for data exploration, addressing key metrics like total orders, sales revenue, and average order quantity. Analytical questions focused on discount rates, revenue distribution, and customer spending patterns. Using data visualization tools, I created charts and graphs to illustrate trends and insights. Finally, I provided actionable recommendations to optimize sales strategies and enhance overall performance.

[BACK TO AGENDA](#)

# Data Exploration

01

**Total number of orders**

SQL QUERY:

```
SELECT COUNT(*) AS total_orders  
FROM sales_data;
```

**25000**

02

**Total sales revenue**

SQL QUERY:

```
SELECT SUM(amount_per_unit *  
ordered_quantity) AS  
total_sales_revenue FROM  
sales_data  
WHERE order_status = 'CLOSED';
```

**88403552.725**

03

**Average order quantity**

SQL QUERY:

```
SELECT  
ROUND(AVG(ordered_quantity),  
2) AS average_order_quantity  
FROM sales_data;
```

**22.84**

[BACK TO AGENDA](#)

04

## Distribution of orders by warehouse and store

### SQL QUERY:

```
SELECT order_warehouse_id, store_id,
COUNT(*) AS order_count
FROM SALES_DATA
GROUP BY order_warehouse_id, store_id;
```

# Data Exploration

Output of first few rows:

Results	Explain	Describe	Saved SQL	History
ORDER_WAREHOUSE_ID	STORE_ID	ORDER_COUNT		
15	5qQncdfXpBQ6cwiFP9nd87	78		
2	FOxy6fUZZUGqy6yqejuUa	71		
9	70rkqCk86iNc8FedKQ4MqP	56		
8	1TrqGzT37l31lsrYN5Kyn8	55		
3	19OiW3FdOOqHa0RnzThMll	53		
2	3gdPnWstu13FnbWvUuzjEn	51		
4	1yCyaUMSjUNaMPA4dBMSCI	51		
2	2nDuQzdVEGeSPB6yzJJf8	51		
9	6GCJ8YtCRtPVFOz7dxhqkI	48		

05



## Top selling items

### SQL QUERY:

```
SELECT DISTINCT item_id,
COUNT(ordered_quantity) AS order_count
FROM SALES_DATA
GROUP BY item_id
ORDER BY order_count DESC;
```

# Data Exploration

## Output of first few rows:

Results	Explain	Describe	Saved SQL	History
ITEM_ID			ORDER_COUNT	
P193053596797216954			693	
P411356364996092377			284	
P510597221483309850			255	
P881311747325487679			254	
P5665770111132			218	
P5904807559324			202	
P257186224944449712			183	
P118574423936666541			182	
P625735472986144560			166	
P668383357090664202			165	
More than 10 rows available. Increase rows selector to view more rows.				
<div><div> hayahahmeduser2@gmail.com</div><div></div><div>Copyright © 1999, 2023, Oracle and/or its</div><div>Oracle APEX 23.2.</div></div>				

BACK TO AGENDA



# Analytical Questions

Q1

What is the total revenue generated by each store?

```
SELECT store_id,
       SUM(amount_per_unit * ordered_quantity -
           item_discount) AS total_revenue
FROM SALES_DATA
WHERE order_status = 'CLOSED'
GROUP BY store_id
ORDER BY total_revenue DESC;
```

STORE_ID	TOTAL_REVENUE
4PdVF4u8WXhO4xnSMCYIzs	2560000
49WcD3UqaC72tOQ6tood3n	2030000
6FFhVY8Ujlw7dpqveE2UTh	1909190
6nEbvZQMYLEkxxN4JwVmEq	1899077
QAJXF0B4KcMUL1Oyibdo	1280515
5fyoaFSI2Qggg56HCGEtIB	1180000
6p39aRrK34DT9A8RWfH5Ck	1180000

Q2

Which warehouse has the highest average order value (total sales revenue divided by number of orders)?

```
SELECT order_warehouse_id,
       ROUND(AVG(amount_per_unit *
           ordered_quantity),2) AS avg_order_value
FROM SALES_DATA
GROUP BY order_warehouse_id
HAVING order_warehouse_id is not NULL
ORDER BY avg_order_value DESC;
```

"The analysis reveals that Warehouse 14 boasts the highest average order value, with an average of **\$76,146.67** per order

BACK TO AGENDA

# Analytical Questions

Q3

What is the overall discount rate (average discount per item sold)?

```
SELECT ROUND(AVG(item_discount /  
(amount_per_unit * ordered_quantity)),2) AS  
avg_discount_rate  
FROM SALES_DATA  
WHERE amount_per_unit > 0 AND ordered_quantity  
> 0 AND item_discount > 0;
```

"The overall discount rate, calculated as the average discount per item sold, is found to be **0.03**."

BACK TO AGENDA

Q4

Identify the top 5 customers (based on total amount spent).

```
SELECT order_number AS top_customers,  
       SUM(amount_per_unit * ordered_quantity) AS  
total_order_value  
FROM SALES_DATA  
GROUP BY order_number  
ORDER BY total_order_value DESC;
```

TOP_CUSTOMERS	TOTAL_ORDER_VALUE
23034138254	160000000
23051225637	3538028
23039497453	3360070
23055942343	2616020
23046539355	2030000



# Analytical Questions

Q5

Calculate the week-over-week growth rate of sales revenue.

```
SELECT TO_CHAR(TO_DATE(order_date, 'DD/MM/YYYY'), 'IW') AS week, SUM(amount_per_unit * ordered_quantity) AS weekly_revenue,  
ROUND( ( SUM(amount_per_unit * ordered_quantity - item_discount) - LAG(SUM(amount_per_unit * ordered_quantity)) OVER (ORDER BY  
TO_DATE(order_date, 'DD/MM/YYYY'))  
/ LAG(SUM(amount_per_unit * ordered_quantity)) OVER (ORDER BY TO_DATE(order_date, 'DD/MM/YYYY')) * 100, 2) AS growth_rate  
FROM SALES_DATA_FINAL  
WHERE TO_DATE(order_date, 'DD/MM/YYYY') >= TO_DATE('01/02/2023', 'DD/MM/YYYY') AND TO_DATE(order_date, 'DD/MM/YYYY') <=  
TO_DATE('28/02/2023', 'DD/MM/YYYY')  
GROUP BY TO_CHAR(TO_DATE(order_date, 'DD/MM/YYYY'), 'IW') ORDER BY TO_CHAR(TO_DATE(order_date, 'DD/MM/YYYY'), 'IW');
```

**Assumption:** The question asks to calculate month-over-month growth but the given dataset is of the month of February, so months can't be compared therefore i've performed week over week growth rate of sales revenue.

[BACK TO AGENDA](#)

# Analytical Questions

Q6

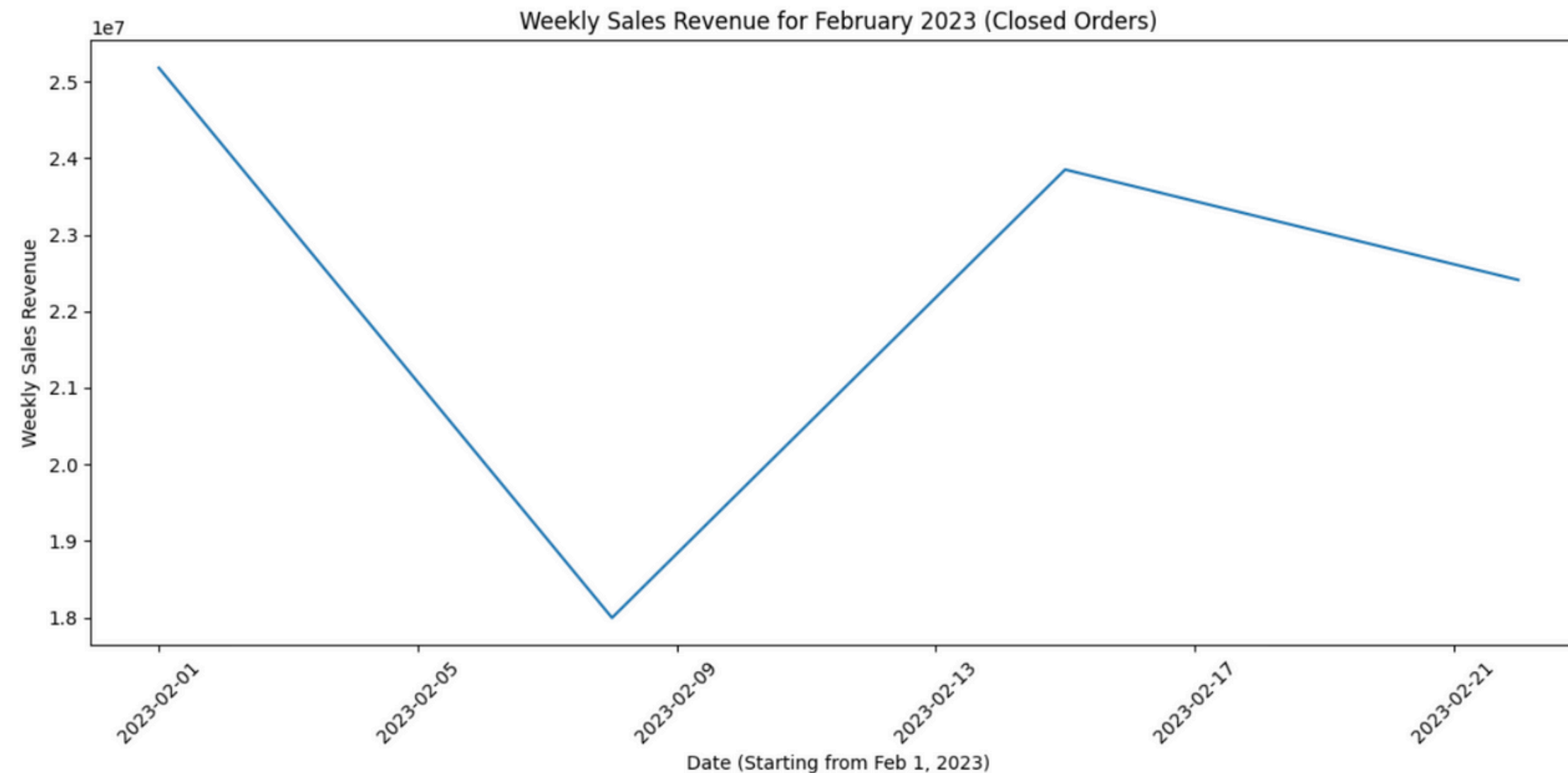
Determine the percentage of orders that were canceled

```
SELECT  
  COUNT(CASE WHEN order_status = 'CANCELLED' THEN 1 ELSE NULL END) AS canceled_orders,  
  COUNT(*) AS total_orders,  
  ROUND((COUNT(CASE WHEN order_status = 'CANCELLED' THEN 1 ELSE NULL END) / COUNT(*)) *  
    100, 2) AS cancellation_rate_percent  
FROM SALES_DATA;
```

CANCELED_ORDERS	TOTAL_ORDERS	CANCELLATION_RATE_PERCENT
1379	25000	5.52

[BACK TO AGENDA](#)

```
Weekly Sales Revenue:
order_date    sales_revenue
2023-02-01    2.518065e+07
2023-02-08    1.799515e+07
2023-02-15    2.385160e+07
2023-02-22    2.240980e+07
Freq: 7D, Name: sales_revenue, dtype: float64
```



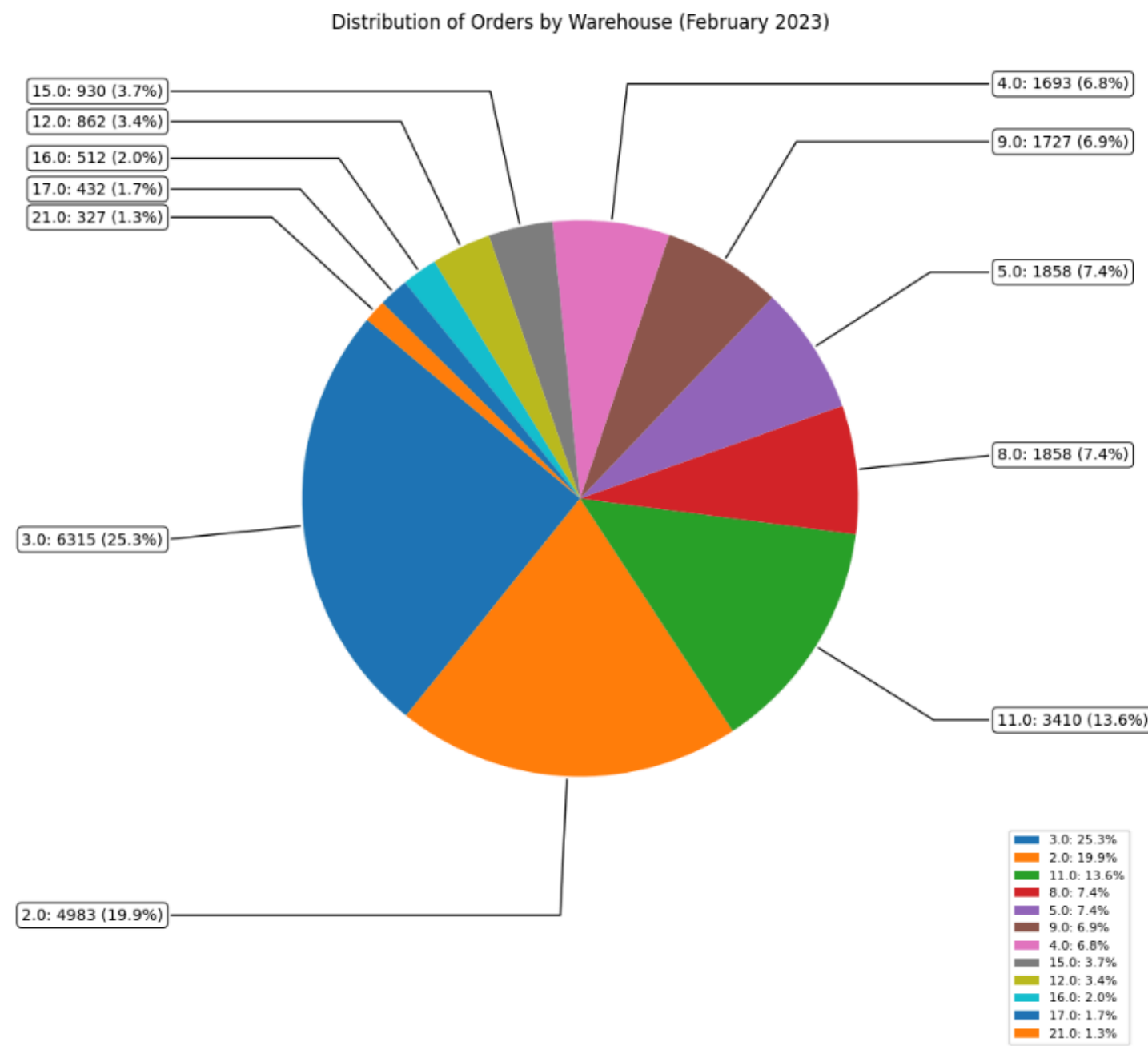
The weekly sales revenue for February 2023 shows a fluctuating pattern rather than a clear trend. There's a dip in revenue during the first week, followed by a significant jump in week two. While week three maintains a similar level of sales to week two, week four dips again.

# بازار BAZAAR

# Sales Revenue Over time Analysis

Python code given in separate file

[BACK TO AGENDA](#)



- **Warehouse 3.0** has the most orders at 6315, which accounts for 25.3% of the total orders.
- **Warehouse 2.0** follows with 4983 orders, which is 19.9% of the total orders.
- **Warehouse 11.0** has 13.6% of the total orders, which is 3410 orders.
- **Warehouse 5.0** and **8.0** each account for 7.4% of the total orders, which is 1858 orders each.
- The remaining **warehouses, 4.0, 9.0, 12.0, 15.0, 16.0, 17.0, and 21.0**, account for a combined total of 18.7% of the total orders.

# بازار BAZAAR

## Distribution of Orders by Warehouse Analysis

Python code given in separate file

[BACK TO AGENDA](#)

# Additional Insights/Findings

01

how much sale is decreased due to order status cancellation

```
SELECT
  SUM(CASE WHEN order_status = 'CLOSED' THEN amount_per_unit *
ordered_quantity - item_discount ELSE 0 END) AS total_sales,
  SUM(CASE WHEN order_status = 'CANCELLED' THEN amount_per_unit *
ordered_quantity - item_discount ELSE 0 END) AS cancelled_sales,
  CASE WHEN SUM(amount_per_unit * ordered_quantity) > 0 THEN
    SUM(CASE WHEN order_status = 'CANCELLED' THEN amount_per_unit *
ordered_quantity - item_discount ELSE 0 END) /
    SUM(amount_per_unit * ordered_quantity) * 100
  ELSE 0 END AS sales_decrease_percent
FROM sales_data;
```

TOTAL_SALES	CANCELLED_SALES
88403552.72459452999933866	23514133.2389994000000003

SALES_DECREASE_PERCENT
8.44424771417101392987276100084488282805

Due to order cancellation the totalsales is  
affected by approx 8.5%

BACK TO AGENDA

# Additional Insights/Findings

02

identify items that customers were least interested in purchasing

```
SELECT item_id, COUNT(DISTINCT order_number) AS purchase_count
FROM sales_data
WHERE order_status = 'CLOSED' -- Consider only completed orders
GROUP BY item_id
ORDER BY purchase_count ASC;
```

ITEM_ID	PURCHASE_COUNT
P446001240529669654	1
P923702885439835165	1
P540042393405185445	1
P606005160052728104	1
P485696651316227589	1
P794866551389069272	1
P945474456419039571	1
P396478733997386649	1
P769693195405979974	1
P5470867652764	1

More than 10 items have just purchase count of 1

[BACK TO AGENDA](#)





# Recommendations for increased sales

Investigate what these stores with the highest total revenue have in common (e.g., location, product mix, marketing strategies).

Analyze stores with lower revenue. Consider targeted promotions or discounts for these stores to boost sales and attract customers. You could offer temporary price reductions, bundle deals, or loyalty program incentives.

Analyze order cancellation data to understand reasons for lost sales and develop strategies to address them.

Identify products frequently purchased together. Recommend complementary products (cross-selling) or higher-value versions (upselling) at checkout or through targeted campaigns.

[BACK TO AGENDA](#)



# Get In Touch



## Email

hayahahmeduser2@gmail.com



## Linkedin

<https://www.linkedin.com/in/hayah-ahmed/>



## Call

+92-3363717227

BACK TO AGENDA