

GROUP 11

**NAME: PRAGATI GUPTA
SAP ID: 590028492**

**NAME: MIMANSHA SINGHAL
SAP ID: 590028676**

PROJECT REPORT

Restaurant Order Management System

1. Problem Statement :

In restaurants, taking orders and managing them manually can be confusing. Sometimes items get missed, wrong bills are made, or it becomes difficult to remember all orders and sales of the day.

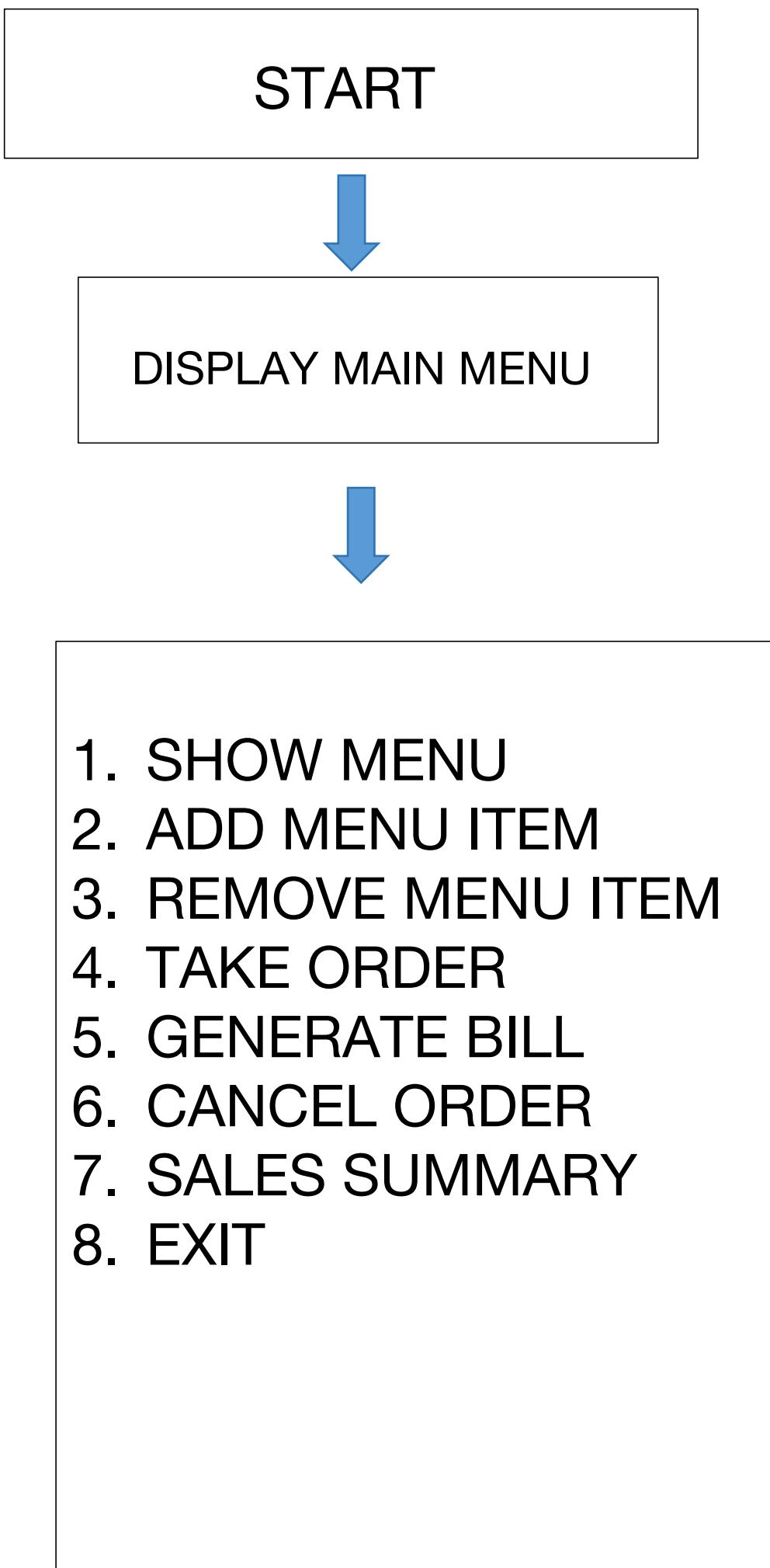
So to solve this problem, we created a **Restaurant Order Management System using Linux Shell Scripting**.

This project helps in:

- # Showing menu items
- # Adding and removing menu items
- # Taking customer orders (multiple items)
- # Generating bills
- # Canceling orders
- # Checking daily sales

It makes restaurant work faster and reduces mistakes.

2. Flowchart :





PERFORM SELECTED TASK



CONTINUE? (y/n)



EXIT

3. Algorithm :

1. Start the program
2. Create required data files (menu, orders, sales)
3. Show the main menu to the user
4. Accept the choice from the user
5. If the choice is:

```
# Show Menu → Display all items  
# Add Menu Item → Take item details and save  
# Remove Menu Item → Delete item by ID  
# Take Order → Take multiple items and calculate total  
# Generate Bill → Show purchased items and final amount  
# Cancel Order → Delete an order by ID  
# Sales Summary → Show total sales of the day  
# Exit → End the program
```

4. Problems We Faced:

1. Bill total was coming as 0

Because the loop was not adding the values correctly.

2. Errors while multiplying price and quantity

Because bc command was missing and caused syntax errors.

3. Order calculation failed when multiple items were added

The total was not updating properly for each item.

4. No option to remove menu items

We had to add delete functionality later.

5. Sometimes item ID was not found and order stopped

We added a check message to show if the menu item does not exist.

6. Formatting of bill was messy in beginning

We fixed it using awk and printf for proper alignment.

5. Snippet of Code:

```
take_order() {
    echo "Enter Order ID:"
    read oid

    while true
    do
        echo "Enter Item ID:"
        read iid
        echo "Enter Quantity:"
        read qty

        price=$(awk -v id="$iid" '$1==id {print $3}' "$MENU_FILE")
        total=$(awk -v p="$price" -v q="$qty" '
BEGIN{print p*q}')

        echo "$oid $iid $qty $total" >> $ORDERS_FILE
        echo "$total" >> $SALES_FILE

        echo "Add another item? (y/n)"
        read choice
        if [ "$choice" != "y" ]; then
            break
        fi
    done

    echo "Order Completed Successfully!"
}
```

6. Output :

```
ubuntu@ubuntu:~$ ./rest.sh
=====
 RESTAURANT ORDER MANAGEMENT SYSTEM
=====
1. Show Menu
2. Add Menu Item
3. Remove Menu Item
4. Take New Order
5. Generate Bill
6. Cancel Order
7. Sales Summary
8. Exit
=====
Enter choice:
█
```

7. Conclusion:

By doing this project, we learned how shell scripting can be used in real life to manage tasks automatically.

We understood how to use loops, conditions, awk, sed, file handling, and menus in bash programming.

This project helped us improve our logical thinking and solve practical problems.

THANK YOU
