PROCEDURAL PROGRAMMING USING C

(DBMS+C) CAPSTONE PROJECT

TOPIC :

SUPERMARKET MANAGEMENT SYSTEM

TEAM 7:

- DIVYASHREE S (CH.SC.U4CSE23109)
- JOSHITHAA R (CH.SC.U4CSE23115)
- RUPA ATCHAYA A S (CH.SC.U4CSE23140)
- SWATHI J B (CH.SC.U4CSE23145)

C CODE OF THE PROJECT:

main.c

```
#include <stdio.h>
#include <stdlib.h>
#include <mysql.h>
#include "customer.h"
#include "inventory.h"
#include "employee.h"
#include "transaction.h"
#include "supplier.h"
#include "reporting.h"

int main() {
    MYSQL *conn = mysql_init(NULL);

    // Connect to the database
    if (conn == NULL) {
        fprintf(stderr, "mysql_init() failed\n");
        return EXIT_FAILURE;
    }

    if (mysql_real_connect(conn, "localhost", "root", "rupa2004",
"AMRITA_SUPERMARKET", 0, NULL, 0) == NULL) {
        fprintf(stderr, "mysql_real_connect() failed: %s\n",
mysql_error(conn));
```

```
        mysql_close(conn);
        return EXIT_FAILURE;
    }

    int choice;
    do {
        printf("\nSupermarket Management System\n");
        printf("1. Add Customer\n");
        printf("2. View Customer\n");
        printf("3. View All Customers\n");
        printf("4. Add Inventory\n");
        printf("5. Add Employee\n");
        printf("6. Update Employee\n");
        printf("7. Delete Employee\n");
        printf("8. View Employee\n");
        printf("9. View All Employees\n");
        printf("10. Add Transaction\n");
        printf("11. View Inventory\n");
        printf("12. View Transactions\n");
        printf("13. Add Supplier\n");
        printf("14. Update Supplier\n");
        printf("15. Delete Supplier\n");
        printf("16. View Suppliers\n");
        printf("17. Sales Analysis\n");  // New menu option for sales
analysis
```

```
        printf("18. Customer Behavior Reports\n");  // New menu option for customer behavior
        printf("0. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1: add_customer(conn); break;
            case 2: view_customer(conn); break;
            case 3: view_all_customers(conn); break;
            case 4: add_inventory(conn); break;
            case 5: add_employee(conn); break;
            case 6: update_employee(conn); break;
            case 7: delete_employee(conn); break;
            case 8: view_employee(conn); break;
            case 9: view_all_employees(conn); break;
            case 10: add_transaction(conn); break;
            case 11: view_inventory(conn); break;
            case 12: view_transactions(conn); break;
            case 13: add_supplier(conn); break;
            case 14: update_supplier(conn); break;
            case 15: delete_supplier(conn); break;
            case 16: view_suppliers(conn); break;
            case 17: sales_analysis(conn); break;  // Call sales analysis function
```

```
            case 18: customer_behavior_report(conn); break;  // Call
customer behavior report function
            case 0: break; // Exit the loop
            default: printf("Invalid choice. Please try again.\n");
        }
    } while (choice != 0);

    mysql_close(conn);
    return EXIT_SUCCESS;
}
```

customer.h

```
// customer.h
#ifndef CUSTOMER_H
#define CUSTOMER_H

#include <mysql.h>

void add_customer(MYSQL *conn);
void update_customer(MYSQL *conn);
void delete_customer(MYSQL *conn);
void view_customer(MYSQL *conn);
void view_all_customers(MYSQL *conn);

#endif // CUSTOMER_H
```

inventory.h

```
// inventory.h
#ifndef INVENTORY_H
#define INVENTORY_H

#include <mysql.h>

void add_inventory(MYSQL *conn);
void view_inventory(MYSQL *conn);

#endif // INVENTORY_H
```

employee.h

```
// employee.h
#ifndef EMPLOYEE_H
#define EMPLOYEE_H

#include <mysql.h>
void add_employee(MYSQL *conn);
void update_employee(MYSQL *conn);
void delete_employee(MYSQL *conn);
void view_employee(MYSQL *conn);
void view_all_employees(MYSQL *conn);
#endif // EMPLOYEE_H
```

transaction.h

```
// transaction.h
#ifndef TRANSACTION_H
#define TRANSACTION_H

#include <mysql.h>

void add_transaction(MYSQL *conn);
void view_transactions(MYSQL *conn);

#endif // TRANSACTION_H
```

supplier.h

```
// supplier.h
#ifndef SUPPLIER_H
#define SUPPLIER_H

#include <mysql.h>
void add_supplier(MYSQL *conn);
void update_supplier(MYSQL *conn);
void delete_supplier(MYSQL *conn);
void view_suppliers(MYSQL *conn);

#endif // SUPPLIER_H
```

reporting.h

```
// reporting.h
#ifndef REPORTING_H
#define REPORTING_H

void sales_analysis(MYSQL *conn);
void customer_behavior_report(MYSQL *conn);

#endif // REPORTING_H
```

customer.c

```
// customer.c
#include <stdio.h>
#include <mysql.h>
#include "customer.h"

void add_customer(MYSQL *conn) {
    char name[50], contact[15], email[50];

    printf("Enter customer name: ");
    scanf("%s", name);
    printf("Enter contact number: ");
    scanf("%s", contact);
    printf("Enter email: ");
```

```
    scanf("%s", email);

    char query[256];
    snprintf(query, sizeof(query),
             "INSERT INTO customers (Name, Contact, Email) VALUES ('%s', '%s', '%s')",
             name, contact, email);

    if (mysql_query(conn, query)) {
        fprintf(stderr, "Error adding customer: %s\n", mysql_error(conn));
    } else {
        printf("Customer added successfully.\n");
    }
}

void update_customer(MYSQL *conn) {
    int customer_id;
    char name[50], contact[15], email[50];

    printf("Enter customer ID to update: ");
    scanf("%d", &customer_id);
    printf("Enter new customer name: ");
    scanf("%s", name);
    printf("Enter new contact number: ");
    scanf("%s", contact);
```

```
    printf("Enter new email: ");
    scanf("%s", email);

    char query[256];
    snprintf(query, sizeof(query),
        "UPDATE customers SET Name='%s', Contact='%s', Email='%s'
WHERE CustomerID=%d",
        name, contact, email, customer_id);

    if (mysql_query(conn, query)) {
        fprintf(stderr, "Error updating customer: %s\n",
mysql_error(conn));
    } else {
        printf("Customer updated successfully.\n");
    }
}

void delete_customer(MYSQL *conn) {
    int customer_id;

    printf("Enter customer ID to delete: ");
    scanf("%d", &customer_id);

    char query[256];
    snprintf(query, sizeof(query),
```

```
        "DELETE FROM customers WHERE CustomerID=%d",
customer_id);

    if (mysql_query(conn, query)) {
        fprintf(stderr, "Error deleting customer: %s\n",
mysql_error(conn));
    } else {
        printf("Customer deleted successfully.\n");
    }
}
void view_customer(MYSQL *conn) {
    int customer_id;

    printf("Enter customer ID to view: ");
    scanf("%d", &customer_id);

    char query[256];
    snprintf(query, sizeof(query),
        "SELECT * FROM customers WHERE CustomerID=%d",
customer_id);

    if (mysql_query(conn, query)) {
        fprintf(stderr, "Error retrieving customer: %s\n",
mysql_error(conn));
        return;
    }
```

```
    MYSQL_RES *res = mysql_store_result(conn);
    MYSQL_ROW row;

    if ((row = mysql_fetch_row(res))) {
        printf("Customer ID: %s\n", row[0]);
        printf("Name: %s\n", row[1]);
        printf("Contact: %s\n", row[2]);
        printf("Email: %s\n", row[3]);
        printf("Loyalty Points: %s\n", row[4]);
    } else {
        printf("No customer found with ID %d.\n", customer_id);
    }

    mysql_free_result(res);
}


void view_all_customers(MYSQL *conn) {
    if (mysql_query(conn, "SELECT * FROM customers")) {
        fprintf(stderr, "Error retrieving customers: %s\n",
mysql_error(conn));
        return;
    }
```

```
    MYSQL_RES *res = mysql_store_result(conn);
    MYSQL_ROW row;

    printf("All Customers:\n");
    while ((row = mysql_fetch_row(res))) {
        printf("Customer ID: %s, Name: %s, Contact: %s, Email: %s,
Loyalty Points: %s\n",
               row[0], row[1], row[2], row[3], row[4]);
    }

    mysql_free_result(res);
}
```

inventory.c

```
// inventory.c
#include <stdio.h>
#include <mysql.h>
#include "inventory.h"

void add_inventory(MYSQL *conn) {
    char product_name[50];
    int quantity;
    float price;

    printf("Enter product name: ");
```

```
    scanf("%s", product_name);
    printf("Enter quantity: ");
    scanf("%d", &quantity);
    printf("Enter price: ");
    scanf("%f", &price);

    char query[256];
    snprintf(query, sizeof(query),
        "INSERT INTO inventory (ProductName, Quantity, Price)
VALUES ('%s', %d, %.2f)",
        product_name, quantity, price);

    if (mysql_query(conn, query)) {
        fprintf(stderr, "Error adding inventory: %s\n",
mysql_error(conn));
    } else {
        printf("Inventory added successfully.\n");
    }
}

void view_inventory(MYSQL *conn) {
    if (mysql_query(conn, "SELECT * FROM inventory")) {
        fprintf(stderr, "Error retrieving inventory: %s\n",
mysql_error(conn));
        return;
    }
```

```
    MYSQL_RES *res = mysql_store_result(conn);
    MYSQL_ROW row;

    printf("Inventory:\n");
    while ((row = mysql_fetch_row(res))) {
        printf("Product ID: %s, Product Name: %s, Quantity: %s, Price:
%s\n",
            row[0], row[1], row[2], row[3]);
    }

    mysql_free_result(res);
}
```

employee.c

```
// employee.c
#include <stdio.h>
#include <mysql.h>
#include "employee.h"

void add_employee(MYSQL *conn) {
    char name[50], position[30];
    float salary;

    printf("Enter employee name: ");
```

```
    scanf("%s", name);
    printf("Enter position: ");
    scanf("%s", position);
    printf("Enter salary: ");
    scanf("%f", &salary);

    char query[256];
    snprintf(query, sizeof(query),
            "INSERT INTO employees (Name, Position, Salary, HireDate)
VALUES ('%s', '%s', %.2f, NOW())",
            name, position, salary);

    if (mysql_query(conn, query)) {
        fprintf(stderr, "Error adding employee: %s\n",
mysql_error(conn));
    } else {
        printf("Employee added successfully.\n");
    }
}

void update_employee(MYSQL *conn) {
    int employee_id;
    char name[50], position[30];
    float salary;

    printf("Enter employee ID to update: ");
```

```
    scanf("%d", &employee_id);
    printf("Enter new employee name: ");
    scanf("%s", name);
    printf("Enter new position: ");
    scanf("%s", position);
    printf("Enter new salary: ");
    scanf("%f", &salary);

    char query[256];
    snprintf(query, sizeof(query),
             "UPDATE employees SET Name='%s', Position='%s',
Salary=%.2f WHERE EmployeeID=%d",
             name, position, salary, employee_id);

    if (mysql_query(conn, query)) {
        fprintf(stderr, "Error updating employee: %s\n",
mysql_error(conn));
    } else {
        printf("Employee updated successfully.\n");
    }
}

void delete_employee(MYSQL *conn) {
    int employee_id;

    printf("Enter employee ID to delete: ");
```

```
    scanf("%d", &employee_id);

    char query[256];
    snprintf(query, sizeof(query),
            "DELETE FROM employees WHERE EmployeeID=%d",
employee_id);

    if (mysql_query(conn, query)) {
        fprintf(stderr, "Error deleting employee: %s\n",
mysql_error(conn));
    } else {
        printf("Employee deleted successfully.\n");
    }
}

void view_employee(MYSQL *conn) {
    int employee_id;

    printf("Enter employee ID to view: ");
    scanf("%d", &employee_id);

    char query[256];
    snprintf(query, sizeof(query),
            "SELECT * FROM employees WHERE EmployeeID=%d",
employee_id);
```

```
    if (mysql_query(conn, query)) {
        fprintf(stderr, "Error retrieving employee: %s\n",
mysql_error(conn));
        return;
    }

    MYSQL_RES *res = mysql_store_result(conn);
    MYSQL_ROW row;

    if ((row = mysql_fetch_row(res))) {
        printf("Employee ID: %s\n", row[0]);
        printf("Name: %s\n", row[1]);
        printf("Position: %s\n", row[2]);
        printf("Salary: %s\n", row[3]);
        printf("Hire Date: %s\n", row[4]);
    } else {
        printf("No employee found with ID %d.\n", employee_id);
    }
        mysql_free_result(res);
}

void view_all_employees(MYSQL *conn) {
    if (mysql_query(conn, "SELECT * FROM employees")) {
        fprintf(stderr, "Error retrieving employees: %s\n",
mysql_error(conn));
        return;
```

```
    }

    MYSQL_RES *res = mysql_store_result(conn);
    MYSQL_ROW row;

    printf("All Employees:\n");
    while ((row = mysql_fetch_row(res))) {
        printf("Employee ID: %s, Name: %s, Position: %s, Salary: %s, Hire
Date: %s\n",
               row[0], row[1], row[2], row[3], row[4]);
    }

    mysql_free_result(res);
}
```

transaction.c

```
// transaction.c
#include <stdio.h>
#include <mysql.h>
#include "transaction.h"

void add_transaction(MYSQL *conn) {
    float amount;
    int customer_id, employee_id;
```

```
    printf("Enter transaction amount: ");
    scanf("%f", &amount);
    printf("Enter customer ID: ");
    scanf("%d", &customer_id);
    printf("Enter employee ID: ");
    scanf("%d", &employee_id);

    char query[256];
    snprintf(query, sizeof(query),
             "INSERT INTO transactions (Date, Amount, CustomerID,
EmployeeID) VALUES (NOW(), %.2f, %d, %d)",
             amount, customer_id, employee_id);

    if (mysql_query(conn, query)) {
        fprintf(stderr, "Error adding transaction: %s\n",
mysql_error(conn));
    } else {
        printf("Transaction added successfully.\n");
    }
}

void view_transactions(MYSQL *conn) {
    if (mysql_query(conn, "SELECT * FROM transactions")) {
        fprintf(stderr, "Error retrieving transactions: %s\n",
mysql_error(conn));
        return;
```

```
    }

    MYSQL_RES *res = mysql_store_result(conn);
    MYSQL_ROW row;

    printf("Transactions:\n");
    while ((row = mysql_fetch_row(res))) {
        printf("Transaction ID: %s, Date: %s, Amount: %s, Customer ID: %s, Employee ID: %s\n",
               row[0], row[1], row[2], row[3], row[4]);
    }

    mysql_free_result(res);
}
```

supplier.c

```
// supplier.c
#include <stdio.h>
#include <mysql.h>
#include "supplier.h"

void add_supplier(MYSQL *conn) {
    char name[50], contact[15], email[50];

    printf("Enter supplier name: ");
```

```
    scanf("%s", name);
    printf("Enter contact number: ");
    scanf("%s", contact);
    printf("Enter email: ");
    scanf("%s", email);

    char query[256];
    snprintf(query, sizeof(query),
             "INSERT INTO suppliers (Name, Contact, Email) VALUES ('%s',
'%s', '%s')",
             name, contact, email);

    if (mysql_query(conn, query)) {
        fprintf(stderr, "Error adding supplier: %s\n", mysql_error(conn));
    } else {
        printf("Supplier added successfully.\n");
    }
}

void update_supplier(MYSQL *conn) {
    int supplier_id;
    char name[50], contact[15], email[50];

    printf("Enter supplier ID to update: ");
    scanf("%d", &supplier_id);
```

```
    printf("Enter new supplier name: ");
    scanf("%s", name);
    printf("Enter new contact number: ");
    scanf("%s", contact);
    printf("Enter new email: ");
    scanf("%s", email);

    char query[256];
    snprintf(query, sizeof(query),
            "UPDATE suppliers SET Name='%s', Contact='%s', Email='%s'
WHERE SupplierID=%d",
            name, contact, email, supplier_id);

    if (mysql_query(conn, query)) {
        fprintf(stderr, "Error updating supplier: %s\n",
mysql_error(conn));
    } else {
        printf("Supplier updated successfully.\n");
    }
}

void delete_supplier(MYSQL *conn) {
    int supplier_id;

    printf("Enter supplier ID to delete: ");
    scanf("%d", &supplier_id);
```

```
    char query[256];
    snprintf(query, sizeof(query),
            "DELETE FROM suppliers WHERE SupplierID=%d",
supplier_id);

    if (mysql_query(conn, query)) {
        fprintf(stderr, "Error deleting supplier: %s\n",
mysql_error(conn));
    } else {
        printf("Supplier deleted successfully.\n");
    }
}

void view_suppliers(MYSQL *conn) {
    if (mysql_query(conn, "SELECT * FROM suppliers")) {
        fprintf(stderr, "Error retrieving suppliers: %s\n",
mysql_error(conn));
        return;
    }

    MYSQL_RES *res = mysql_store_result(conn);
    MYSQL_ROW row;

    printf("Suppliers:\n");
    while ((row = mysql_fetch_row(res))) {
```

```
        printf("Supplier ID: %s, Name: %s, Contact: %s, Email: %s\n",
            row[0], row[1], row[2], row[3]);
    }

    mysql_free_result(res);
}
```

reporting.c

```
// reporting.c
#include <stdio.h>
#include <mysql.h>
#include "reporting.h"

    void sales_analysis(MYSQL *conn) {
    // SQL query to analyze sales
    const char *query = "SELECT i.ProductName, SUM(t.Amount) as
TotalSales "
                        "FROM Transactions t "
                        "JOIN Inventory i ON t.ProductID = i.ProductID "
                        "GROUP BY i.ProductName "
                        "ORDER BY TotalSales DESC;";

    if (mysql_query(conn, query)) {
        fprintf(stderr, "Query failed: %s\n", mysql_error(conn));
        return;
```

```
    }

    MYSQL_RES *result = mysql_store_result(conn);
    if (result == NULL) {
        fprintf(stderr, "mysql_store_result() failed: %s\n",
mysql_error(conn));
        return;
    }

    printf("\nSales Analysis:\n");
    printf("Product Name | Total Sales\n");
    printf("-------------|-------------\n");
    MYSQL_ROW row;
    while ((row = mysql_fetch_row(result))) {
        printf("%-12s | %s\n", row[0], row[1]);
    }

    mysql_free_result(result);
}

void customer_behavior_report(MYSQL *conn) {
    // Sample SQL query for customer behavior report
    const char *query = "SELECT CustomerID, COUNT(*) AS
PurchaseCount FROM transactions GROUP BY CustomerID";

    if (mysql_query(conn, query)) {
```

```
        fprintf(stderr, "Query failed: %s\n", mysql_error(conn));
        return;
    }

    MYSQL_RES *result = mysql_store_result(conn);
    if (result == NULL) {
        fprintf(stderr, "Could not retrieve result set: %s\n",
mysql_error(conn));
        return;
    }

    // Process the results
    MYSQL_ROW row;
    while ((row = mysql_fetch_row(result))) {
        printf("Customer ID: %s, Purchases: %s\n", row[0], row[1]);
    }

    mysql_free_result(result);
}
```