RESTAURANT MANAGEMENT SYSTEM

PROJECT BASED LEARNING REPORT

Submitted in partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology

In

COMPUTER SCIENCE AND ENGINEERING

By

BATCH-2B

MUDUNOORI SRI CHAITANYA VARMA	**22331A05A5**
MADETI DHANYA SRI	**22331A0592**
PATHIVADA NIKHITA	**22331A05D1**
KOTA JAGADEESH	**22331A0582**

Under the supervision of

Mrs. N. Aswani

Assistant professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MVGR COLLEGE OF ENGINEERING (Autonomous)
VIZIANAGARAM-535005, AP (INDIA)
(Accredited by NBA, NAAC, and Permanently Affiliated to Jawaharlal Nehru Technological University Kakinada)

OCTOBER, 2024

CERTIFICATE

This is to certify that the project report entitled "**RESTAURANT MANAGEMENT SYSTEM**" is being submitted by **MUDUNOORI SRI CHAITANYA VARMA, MADETI DHANYA SRI, PATHIVADA NIKHITA, KOTA JAGADEESH** bearing registered numbers **22331A05A5, 22331A0592, 22331A05D1, 22331A0582** respectively, in partial fulfilment for the award of the degree of "**Bachelor of Technology" in Computer Science and Engineering** is a record of bonafide work done by them under my supervision during the academic year 2024-2025.

Dr. T. Pavan Kumar

Head of the Department,

Department of CSE,
MVGR College of Engineering,
Vizianagaram.

Mrs. N. Aswani

Assistant Professor,

Department of CSE,
MVGR College of Engineering,
Vizianagaram.

ACKNOWLEDGEMENTS

We place on record and warmly acknowledge the continuous encouragement, invaluable supervision, timely suggestions and inspired guidance offered by our guide **Mrs. N. Aswani ,**Asst. Professor**,** Department of Computer Science and Engineering MVGR College of Engineering in bringing this report to a successful completion.

We consider it our privilege to express our deepest gratitude to **Dr T. Pavan Kumar,** Head of the Department for his valuable suggestions and constant motivation that greatly helped the project work to get successfully completed.

We sincerely thank all the members of the staff in the Department of Computer Science & Engineering for their sustained help in our pursuits. We thank all those who contributed directly or indirectly in successfully carrying out this work.

TABLE OF CONTENTS

ABSTRACT

The demand for Restaurant Management Systems (RMS) is rising as restaurants seek greater efficiency, improved customer experiences, and digital solutions. RMS automates key processes like order management, billing, and inventory control, reducing errors and speeding up service. The growth of online ordering and delivery has amplified the need for systems that seamlessly handle both in-house and external orders. Modern consumers expect personalised service, and RMS with integrated Customer Relationship Management (CRM) helps restaurants store customer data, enhance loyalty, and make data-driven decisions. The shift to cloud-based and mobile solutions, accelerated by the pandemic, has made RMS vital for optimising operations, controlling costs, and staying competitive.

The primary objective of a Restaurant Management System (RMS) is to improve operational efficiency by streamlining and automating key processes such as order management, billing, and inventory control. This helps reduce manual errors, speed up service, and ensure smoother daily operations. The secondary objective is to enhance the customer experience by utilising integrated CRM tools to store customer data, track preferences, and offer personalised promotions. This allows restaurants to provide tailored service, build customer loyalty, and increase overall satisfaction. Together, these objectives make RMS a critical tool for restaurant success.

PROBLEM STATEMENT

Restaurants today face multiple challenges, including inefficient order management, billing, and inventory control, leading to delays and customer dissatisfaction. The rise of online ordering and delivery has added complexity, as many businesses struggle to integrate these services with their in-house operations. Customers also demand personalised experiences, but restaurants often lack the tools to effectively manage and utilise customer data. Additionally, poor inventory control results in food waste and increased costs. Without real-time reporting and data insights, restaurants find it difficult to optimise their operations, especially in the face of growing competition and the need for contactless solutions.

How can we develop an efficient Restaurant Management System using a Relational DBMS that meets the needs of all users?

IDENTIFYING ENTITIES , ATTRIBUTES AND RELATIONS

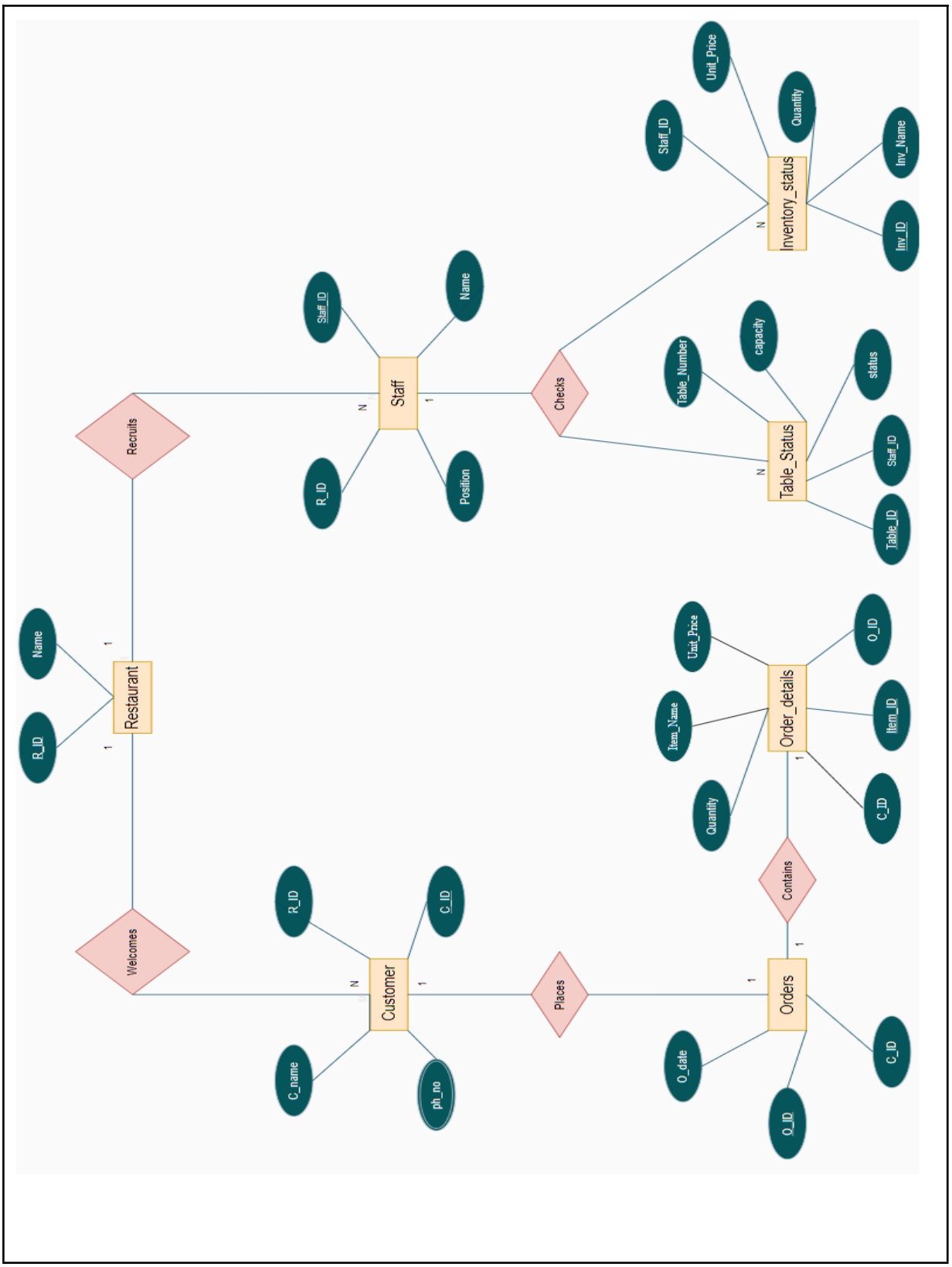
List of Entities and Attributes:

1. **Restaurant**
   - R_ID
   - NAME
2. **Customer**
   - C_ID
   - R_ID
   - C_NAME
   - PH_NO
3. **Orders**
   - O_ID
   - C_ID
   - O_DATE
4. **Order_dtls**
   - C_ID
   - O_ID
   - ITEM_ID
   - ITEM_NAME
   - UNIT_PRICE
   - QUANTITY
5. **Staff**
   - STAFF_ID
   - R_ID
   - NAME
   - POSITION
6. **Table_status**
   - TABLE_ID
   - STAFF_ID
   - TABLE_NUMBER
   - CAPACITY
   - STATUS
7. **Inventory_status**
   - INV_ID
   - STAFF_ID
   - INV_NAME
   - QUANTITY
   - UNIT_PRICE

List Of Relationships:

1. Restaurant Welcomes Customer(one to many)
2. Customer Places Orders (one to one)
3. Order Contains Order_dtls (one to one)
4. Restaurant Recruits Staff (One to many)
5. Staff Checks Table_Status (one to many)
6. Staff Checks Inventory_status (one to many)

ER DIAGRAM


Restaurant
Name
R_ID
Recruits
Staff
Staff_ID
Name
R_ID
Position
Checks
Inventory_status
Unit_Price
Staff_ID
Quantity
Inv_Name
Inv_ID
Table_Status
Table_Number
capacity
status
Staff_ID
Table_ID
Welcomes
Customer
R_ID
C_ID
C_name
ph_no
Places
Orders
O_date
O_ID
C_ID
Contains
Order_details
Unit_Price
Item_Name
Quantity
O_ID
Item_ID
C_ID

IMPLEMENTATION

Database Creation:

1.Table Restaurant

```
CREATE TABLE Restaurant(
   R_id int primary key,
   Name varchar(40)) NOT NULL;

INSERT INTO Restaurant values(101,'House of Biriyanis');

SELECT * FROM Restaurant;
```

R_ID	NAME
101	House of Biriyanis

2. Table Customer

```
CREATE TABLE Customer(
   C_id int primary key,
   R_id int,
   C_name varchar(30) not null,
   ph_no  int ,
   foreign key(R_id) references Restaurant(R_id));

INSERT INTO Customer values(23,101,'Chaitanya',8096714333);
INSERT INTO Customer values(56,101,'Dhanyasri',9876714903);
INSERT INTO Customer values(89,101,'Nikhita',9876500981);
INSERT INTO Customer values(91,101,'Jagadeesh',7245009872);

SELECT * FROM customer;
```

C_ID	R_ID	C_NAME	PH_NO
23	101	Chaitanya	8096714333
56	101	Dhanyasri	9876714903
89	101	Nikhita	9876500981
91	101	Jagadeesh	7245009872

Table Orders

```
CREATE TABLE Orders(
    O_id int primary key,
    C_id int,
    O_date date,
    foreign key(C_id) references Customer(C_id));

INSERT INTO Orders values(1,23,'24-JUNE-24');
INSERT INTO Orders values(2,56,'24-JUNE-24');
INSERT INTO Orders values(3,89,'24-JUNE-24');
INSERT INTO Orders values(4,91,'24-JUNE-24');

SELECT * FROM  Orders;
```

O_ID	C_ID	O_DATE
1	23	24-JUN-24
2	56	24-JUN-24
3	89	24-JUN-24
4	91	24-JUN-24

Table Order_dtls

```
CREATE TABLE Order_details(
   C_id          int,
   O_id          int,
   Item_id       int,
   Item_name     varchar(30),
   Unit_price    int ,
   Quantity      int check(Quantity >0),
   Status        varchar(20),
   PRIMARY KEY (Item_id, O_id),
   FOREIGN KEY (C_id) REFERENCES Customer(C_id),
   FOREIGN KEY (O_id) REFERENCES Orders(O_id)
);
INSERT INTO Order_details values(23,1,2233,'Veg Biriyani',239,2,'Not served');
INSERT INTO Order_details values(23,1,2234,'Chicken Biriyani',289,1,'Not served');
INSERT INTO Order_details values(56,2,2234,'Chicken Biriyani',289,2,' Not served');
INSERT INTO Order_details values(89,3,2244,'Prawns Biriyani',319,1,'Not served');
INSERT INTO Order_details values(89,3,2233,'Veg Biriyani',239,1,'Not served');
INSERT INTO Order_details values(91,4,2255,'Fish Biriyani',329,2,'Not served');

SELECT * FROM  Order_details;
```

C_ID	O_ID	ITEM_ID	ITEM_NAME	UNIT_PRICE	QUANTITY	STATUS
23	1	2233	Veg Biriyani	239	2	Not served
23	1	2234	Chicken Biriyani	289	1	Not served
56	2	2234	Chicken Biriyani	289	2	Not served
89	3	2244	Prawns Biriyani	319	1	Not served
89	3	2233	Veg Biriyani	239	1	Not served
91	4	2255	Fish Biriyani	329	2	Not served

Table Staff:

```
CREATE TABLE Staff (
   Staff_ID INT PRIMARY KEY,
   R_id     INT,
   Name     VARCHAR(25),
   Position VARCHAR(25),
   FOREIGN KEY(R_id) REFERENCES Restaurant(R_id)
);

INSERT INTO Staff values(25,101,'Bhuvana','Manager');
INSERT INTO Staff values(52,101,'Renuka','Cashier');
INSERT INTO Staff values(67,101,'Manoj','Waiter');


SELECT * FROM  Staff;
```

STAFF_ID	R_ID	NAME	POSITION
25	101	Bhuvana	Manager
52	101	Renuka	Cashier
67	101	Manoj	Waiter

Table Table_status:

```
CREATE TABLE Table_status (
   Table_ID        INT   PRIMARY KEY,
   Staff_ID        INT,
   Table_Number    INT,
   Capacity        INT,
   Status          VARCHAR(20),
   FOREIGN KEY(Staff_id) REFERENCES Staff(Staff_id)
);
INSERT INTO Table_status values(1,67,1,4,'Vacant');
INSERT INTO Table_status values(2,67,2,6,'Vacant');
INSERT INTO Table_status values(3,67,3,7,'Occupied');

SELECT * FROM  Table_status;
```

TABLE_ID	STAFF_ID	TABLE_NUMBER	CAPACITY	STATUS
1	67	1	4	Vacant
2	67	2	6	Vacant
3	67	3	7	Occupied

Table Inventory_status:

```
CREATE TABLE Inventory_status(
  Inv_ID INT PRIMARY KEY,
  staff_ID INT,
  Inv_Name VARCHAR(255),
  Quantity INT,
  Unit_Price DECIMAL(10, 2),
  FOREIGN KEY (staff_ID) REFERENCES Staff(Staff_ID)
);

INSERT INTO Inventory_status values(676,25,'Chicken Biriyani',6,289);
INSERT INTO Inventory_status values(125,25,'Veg Biriyani',10,239);
INSERT INTO Inventory_status values(677,25,'Prawn Biriyani',13,319);
INSERT INTO Inventory_status values(178,25,'Fish Biriyani',2,329);

SELECT *  FROM   Inventory_status;
```

INV_ID	STAFF_ID	INV_NAME	QUANTITY	UNIT_PRICE
676	25	Chicken Biriyani	6	289
125	25	Veg Biriyani	10	239
677	25	Prawn Biriyani	13	319
178	25	Fish Biriyani	2	329

Queries

1. *Identify the customers who have ordered 'Chicken Biriyani' and serve their orders if sufficient stock is available without serving partial orders .*

PL/SQL COMMAND:

```
DECLARE
        c NUMBER;
        d NUMBER;
BEGIN
SELECT Quantity INTO c FROM Inventory_status WHERE Inv_name = 'Chicken Biriyani';
SELECT sum(Quantity) INTO d FROM Order_detailss WHERE Item_name='Chicken Biriyani';
        IF c >= d THEN
                DBMS_OUTPUT.PUT_LINE('Order served');

                UPDATE Inventory_status
                SET Quantity = Quantity - d
                WHERE Inv_name = 'Chicken Biriyani';

                UPDATE Order_detailss
                SET status='served'
                WHERE Item_name='Chicken Biriyani';
        ELSE
                DBMS_OUTPUT.PUT_LINE('Out of stock');
        END IF;
END;
```

OUTPUT:

```
Statement processed.
Order served
```

```
SELECT *  FROM   Inventory_status WHERE Inv_name='Chicken Biriyani';
```

INV_ID	STAFF_ID	INV_NAME	QUANTITY	UNIT_PRICE
676	25	Chicken Biriyani	3	289

```
SELECT *  FROM   Order_details WHERE Item_name='Chicken Biriyani';
```

C_ID	O_ID	ITEM_ID	ITEM_NAME	UNIT_PRICE	QUANTITY	STATUS
23	1	2234	Chicken Biriyani	289	1	served
56	2	2234	Chicken Biriyani	289	2	served

2. *Write a query to determine if a table is available to seat a group of 4 customers arriving at the same time.*

SQL COMMAND:

```
SELECT *
FROM Table_status
WHERE capacity>=4
AND
status='Vacant';

UPDATE Table_status
SET Status='Occupied'
WHERE Table_Number=1;
```

OUTPUT:

Table_Number 1 and 2 are vacant with capacity >=4 .

TABLE_ID	STAFF_ID	TABLE_NUMBER	CAPACITY	STATUS
1	67	1	4	Vacant
2	67	2	6	Vacant

Waiter allocates Table_Number 1 to the group of 4.

SELECT * FROM Table_status;

TABLE_ID	STAFF_ID	TABLE_NUMBER	CAPACITY	STATUS
1	67	1	4	Occupied
2	67	2	6	Vacant
3	67	3	7	Occupied

3. *Write a query to retrieve the details of customers who have ordered more than one item.*

SQL COMMAND:

```
SELECT DISTINCT C_ID,C_NAME,PH_NO
FROM Customer NATURAL JOIN  Order_details
WHERE C_ID IN (
      SELECT C_ID
      FROM order_details
      GROUP BY C_ID
      HAVING COUNT(C_ID)>1
      );
```

OUTPUT;

C_ID	C_NAME	PH_NO
89	Nikhita	9876500981
23	Chaitanya	8096714333

4. *Write a query to identify the roles of staff members who manage Inventory and those who manage Tables.*

SQL COMMAND:

```
SELECT STAFF_ID,NAME,POSITION
FROM Staff
WHERE staff_id=(
      SELECT DISTINCT staff_ID
      FROM Inventory_status
      );

SELECT STAFF_ID,NAME,POSITION
FROM Staff
WHERE staff_id=(
      SELECT DISTINCT staff_ID
      FROM Table_status
      );
```

OUTPUT:

Role of Staff Member who manages Inventory:

STAFF_ID	NAME	POSITION
25	Bhuvana	Manager

Role of Staff member Who manages Tables:

STAFF_ID	NAME	POSITION
67	Manoj	Waiter

5. *Write a query to calculate the total sales from orders that have been served.*

SQL COMMAND:

```
SELECT SUM(UNIT_PRICE*QUANTITY) AS Total_sale
FROM Order_detailss
WHERE status='served';
```

OUTPUT:

TOTAL_SALE
867

SUMMARY

This project aims to develop a robust and efficient database tailored specifically for a Restaurant Management System. The primary objective is to design a comprehensive database architecture capable of managing various aspects of restaurant operations, including menu details, table reservations, staff management, and customer profiles. By incorporating modern data management techniques and focusing on data security and integrity, the system is designed to enhance the overall dining experience for customers. The optimised database structure improves operational efficiency, enabling seamless order management, inventory control, and customer interactions, ultimately contributing to enhanced customer satisfaction and streamlined restaurant operations.