**SVKM’s NMIMS**

**School of Technology Management & Engineering, Chandigarh**

A.Y. 2023 - 24

**Course: Database Management Systems**

**Project Report**

|  |  |  |
| --- | --- | --- |
| Program | MBA TECH | |
| Semester | 4 | |
| Name of the Project: | GROCERY MANAGEMENT | |
|  | | |
| Details of Project Members |  |  |
| Batch | Roll No. | Name |
| 2 | A196 | SHRAVANI DHORE |
| 2 | A182 | DIVYANSH CHANDRA |
|  |  |  |
| Date of Submission: | | |

**Contribution of each project Members:**

|  |  |  |
| --- | --- | --- |
| Roll No. | Name: | Contribution |
| A196 | SHRAVANI DHORE | ER MODEL, CODE,10-ENTITIES, RELATIONAL DIAGRAM, |
| A182 | DIVYANSH CHANDRA | NORMALIZATION, 10-ENTITIES,CODE,COMPONENTS |

**Github link of your project:**

**Note:**

1. Create a readme file if you have multiple files
2. All files must be properly named (Example:R004\_DBMSProject)
3. Submit all relevant files of your work ( Report, all SQL files, Any other files)
4. **Plagiarism is highly discouraged (Your report will be checked for plagiarism)**

**Rubrics for the Project evaluation:**

|  |  |
| --- | --- |
| First phase of evaluation:  Innovative Ideas (5 Marks)  Design and Partial implementation (5 Marks) | 10 marks |
| Final phase of evaluation  Implementation, presentation and viva, Self-Learning and Learning Beyond classroom | 10 marks |

**Project Report**

**GROCERY MANAGEMENT**

**by**

**SHRAVANI DHORE, A196**

**DIVYANSH CHANDRA, A182**

**Course: DBMS**

**AY: 2023-24**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Sr no.** | **Topic** | **Page no.** |
| **1** | Storyline |  |
| **2** | Components of Database Design |  |
| **3** | Entity Relationship Diagram |  |
| **4** | Relational Model |  |
| **5** | Normalization |  |
| **6** | SQL Queries |  |
| **7** | Learning from the Project |  |
| **8** | Project Demonstration |  |
| **9** | Self-learning beyond classroom |  |
| **10** | Learning from the project |  |
| **8** | Challenges faced |  |
| **9** | Conclusion |  |

**I. Storyline**

This section should describe the requirements for the chosen database topic. Form a storyline and describe in detail.

**II. Components of Database Design**

Describe all entities along with their attributes here. Also, mention the primary keys for each entity.

Describe all relationships among various entities. Also, specify the cardinality and participation for all relationships.

* **CUSTOMER TABLE-**

CREATE TABLE Customers (customer\_id INT PRIMARY KEY, name VARCHAR(100), email VARCHAR(255), phone\_number VARCHAR(20), address\_id INT, FOREIGN KEY (address\_id) REFERENCES Addresses(address\_id) );

INSERT INTO Customers (customer\_id, name, email, phone\_number, address\_id)

VALUES

(101, 'John Smith', 'john.smith@example.com', '+1 (555) 123-4567', 201),

(102, 'Sarah Johnson', 'sarah.johnson@emailprovider.com', '+1 (555) 987-6543', 202),

(103, 'Michael Lee', 'mlee@gmail.com', '+1 (555) 222-3333', 203),

(104, 'Emily Chen', 'emily\_chen@email.com', '+1 (555) 777-8888', 204),

(105, 'Maria Rodriguez', 'marodriguez@emailprovider.com', '+1 (555) 444-5555', 205);

* **PRODUCT TABLE-**

CREATE TABLE Product (

product\_id INT PRIMARY KEY,

product\_name VARCHAR(255),

stock INT,

category\_id INT,

MRP DECIMAL(10, 2),

Brand VARCHAR(100)

);

INSERT INTO Product (product\_id, product\_name, stock, category\_id, MRP, Brand)

VALUES

(501, 'Laptop X1', 10, 701, 1299.99, 'TechMaster'),

(502, 'Smartphone S2', 20, 702, 899.00, 'ElectroTech'),

(503, 'TV T3', 5, 704, 1499.99, 'ViewSonic'),

(504, 'Earbuds E4', 30, 706, 249.00, 'AudioTech'),

(505, 'Running Shoes R5', 15, 707, 199.95, 'FitFoot');

* **ORDER TABLE-**

CREATE TABLE Orders (

order\_id INT PRIMARY KEY,

shipping\_date DATE,

cart\_id INT,

order\_amount DECIMAL(10, 2),

order\_date DATE

);

INSERT INTO Orders (order\_id, shipping\_date, cart\_id, order\_amount, order\_date)

VALUES

(1001, '2024-03-20', 501, 1299.99, '2024-03-15'),

(1002, '2024-03-21', 502, 899.00, '2024-03-16'),

(1003, '2024-03-22', 503, 1499.99, '2024-03-17'),

(1004, '2024-03-23', 504, 498.00, '2024-03-18'),

(1005, '2024-03-24', 505, 199.95, '2024-03-19');

* **ORDER ITEM TABLE-**

CREATE TABLE OrderItem (

order\_item\_id INT PRIMARY KEY,

order\_id INT,

product\_id INT,

quantity INT,

unit\_price DECIMAL(10, 2),

total\_price DECIMAL(10, 2)

);

INSERT INTO order\_item (shipping\_date, order\_id, cart\_id, order\_amount, order\_date)

VALUES

('2024-03-15', 101, 201, 49.99, '2024-03-10'),

('2024-03-16', 102, 202, 29.99, '2024-03-11'),

('2024-03-17', 103, 203, 99.99, '2024-03-12'),

('2024-03-18', 104, 204, 79.99, '2024-03-13'),

('2024-03-19', 105, 205, 39.99, '2024-03-14');

* **PAYMENT TABLE-**

CREATE TABLE payment (

customer\_id INT,

order\_id INT,

payment\_id INT,

payment\_mode VARCHAR(50),

payment\_date DATE

);

INSERT INTO payment (customer\_id, order\_id, payment\_id, payment\_mode, payment\_date)

VALUES

(101, 201, 301, 'Credit Card', '2024-03-15'),

(102, 202, 302, 'PayPal', '2024-03-16'),

(103, 203, 303, 'Cash On Delivery', '2024-03-17'),

(104, 204, 304, 'Debit Card', '2024-03-18'),

(105, 205, 305, 'Credit Card', '2024-03-19');

* **PAYMENT HISTORY TABLE-**

CREATE TABLE payment\_history (

payment\_history\_id INT,

payment\_id INT,

data VARCHAR(255),

status VARCHAR(50),

transaction\_id VARCHAR(100)

);

INSERT INTO payment\_history (payment\_history\_id, payment\_id, data, status, transaction\_id)

VALUES

(1, 301, 'Payment successful', 'Completed', 'TRX001'),

(2, 302, 'Payment pending verification', 'Pending', 'TRX002'),

(3, 303, 'Payment failed', 'Failed', 'TRX003'),

(4, 304, 'Payment refunded', 'Refunded', 'TRX004'),

(5, 305, 'Payment successful', 'Completed', 'TRX005');

* **ADMINS TABLE-**

CREATE TABLE Admins ( admin\_id INT PRIMARY KEY, username VARCHAR(50), password VARCHAR(255), email VARCHAR(255) );

INSERT INTO Admins (admin\_id, username, password, email, last\_login\_date)

VALUES

(1, 'admin1', 'hashed\_password\_1', 'admin1@example.com'),

(2, 'admin2', 'hashed\_password\_2', 'admin2@example.com'),

(3, 'admin3', 'hashed\_password\_3', 'admin3@example.com'),

(4, 'admin4', 'hashed\_password\_4', 'admin4@example.com'),

(5, 'admin5', 'hashed\_password\_5', 'admin5@example.com');

* **CART TABLE-**

CREATE TABLE cart (

cart\_id INT,

total\_item INT,

grand\_total DECIMAL(10, 2),

customer\_id INT

);

INSERT INTO cart (cart\_id, total\_item, grand\_total, customer\_id)

VALUES

(201, 3, 129.97, 101),

(202, 2, 59.98, 102),

(203, 1, 99.99, 103),

(204, 2, 159.98, 104),

(205, 4, 189.96, 105);

* **SUPPLIER TABLE-**

CREATE TABLE Supplier ( name VARCHAR(100), email VARCHAR(255), phone\_number VARCHAR(20), address VARCHAR(255) );

INSERT INTO Supplier (supplier\_id, name, email, phone\_number, address)

VALUES

(701, 'TechSupply Inc.', 'info@techsupply.com', '+1 (555) 123-4567', '123 Tech Ave, Tech City'),

(702, 'ElectroWorld Suppliers', 'contact@electroworld.com', '+1 (555) 987-6543', '456 Circuit St, Electronics Town'),

(703, 'Gadget Source LLC', 'sales@gadgetsource.com', '+1 (555) 222-3333', '789 Innovation Blvd, Gadget Valley'),

(704, 'Fashion Fabrics Ltd.', 'info@fashionfabrics.com', '+1 (555) 777-8888', '321 Style Ave, Fashion District'),

(705, 'Sporting Gear Co.', 'sales@sportinggear.com', '+1 (555) 444-5555', '555 Fitness Plaza, Sportstown');

* **REVIEW TABLE-**

CREATE TABLE review (

review\_id INT,

description VARCHAR(255),

ratings DECIMAL(3, 2),

product\_id INT,

customer\_id INT

);

INSERT INTO review (review\_id, description, ratings, product\_id, customer\_id)

VALUES

(1, 'Great product, very satisfied with the quality!', 4.5, 101, 201),

(2, 'Fast shipping, item as described.', 4.0, 102, 202),

(3, 'Good value for the price, would recommend.', 3.8, 103, 203),

(4, 'Excellent customer service, resolved my issue quickly.', 5.0, 104, 204),

* ***WISHLIST TABLE-***

CREATE TABLE wishlist (

wishlist\_id INT,

customer\_id INT,

product\_id INT,

added\_at TIMESTAMP

);

INSERT INTO wishlist (wishlist\_id, customer\_id, product\_id, added\_at)

VALUES

(1, 101, 201, '2024-03-15 10:00:00'),

(2, 102, 202, '2024-03-16 11:30:00'),

(3, 103, 203, '2024-03-17 09:45:00'),

(4, 104, 204, '2024-03-18 14:20:00'),

(5, 105, 205, '2024-03-19 16:55:00');

* **SHIPMENT TABLE-**

CREATE TABLE shipment (

shipment\_id INT,

arrival\_date DATE,

estimate\_date DATE,

shipment\_date DATE,

order\_id INT

);

INSERT INTO shipment (shipment\_id, arrival\_date, estimate\_date, shipment\_date, order\_id)

VALUES

(1, '2024-03-20', '2024-03-25', '2024-03-18', 101),

(2, '2024-03-21', '2024-03-26', '2024-03-19', 102),

(3, '2024-03-22', '2024-03-27', '2024-03-20', 103),

(4, '2024-03-23', '2024-03-28', '2024-03-21', 104),

(5, '2024-03-24', '2024-03-29', '2024-03-22', 105);

* **CATEGORY TABLE-**

CREATE TABLE category (

category\_id INT,

category\_name VARCHAR(50),

description VARCHAR(255)

);

INSERT INTO category (category\_id, category\_name, description)

VALUES

(1, 'Electronics', 'Includes all electronic devices such as smartphones, laptops, and cameras.'),

(2, 'Clothing', 'A wide range of clothing items for men, women, and kids.'),

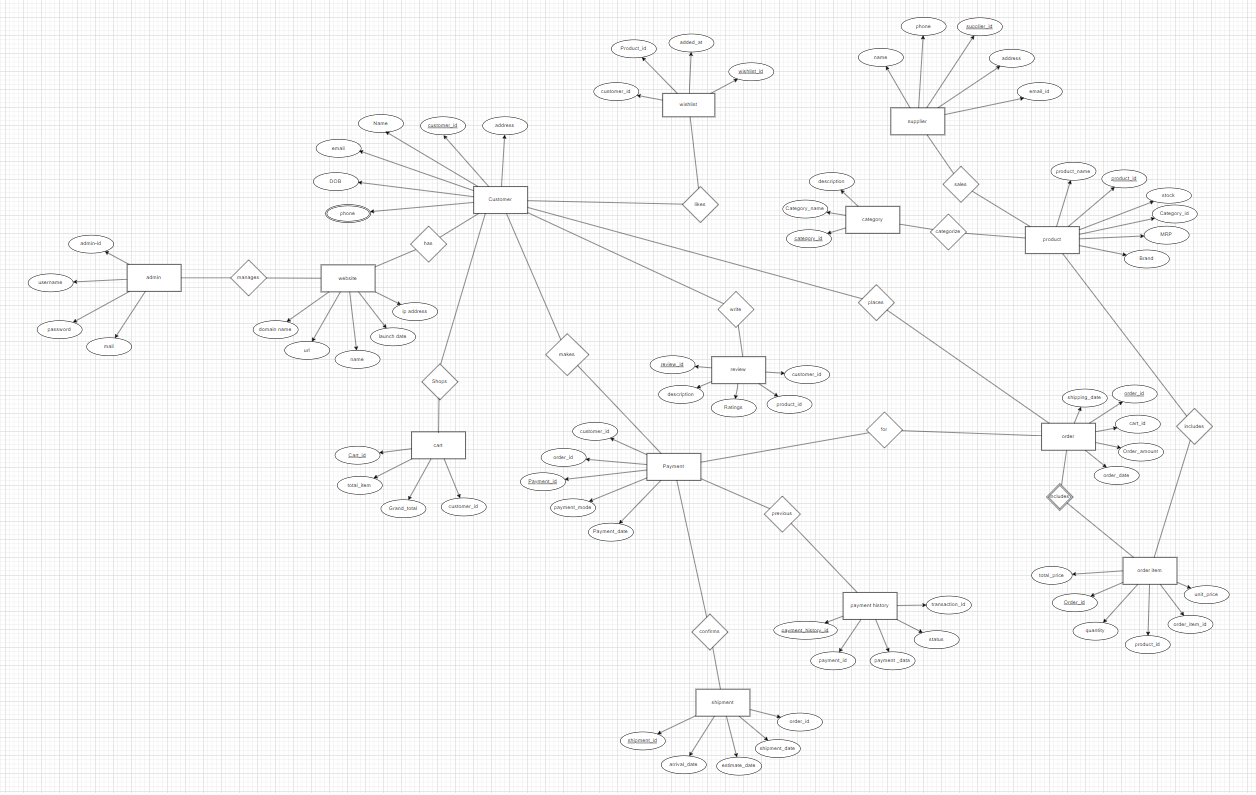
(3, 'Books', 'Various genres of books including fiction, non-fiction, and educational.'),

(4, 'Home & Kitchen', 'Products for home decor, kitchen appliances, and accessories.'),

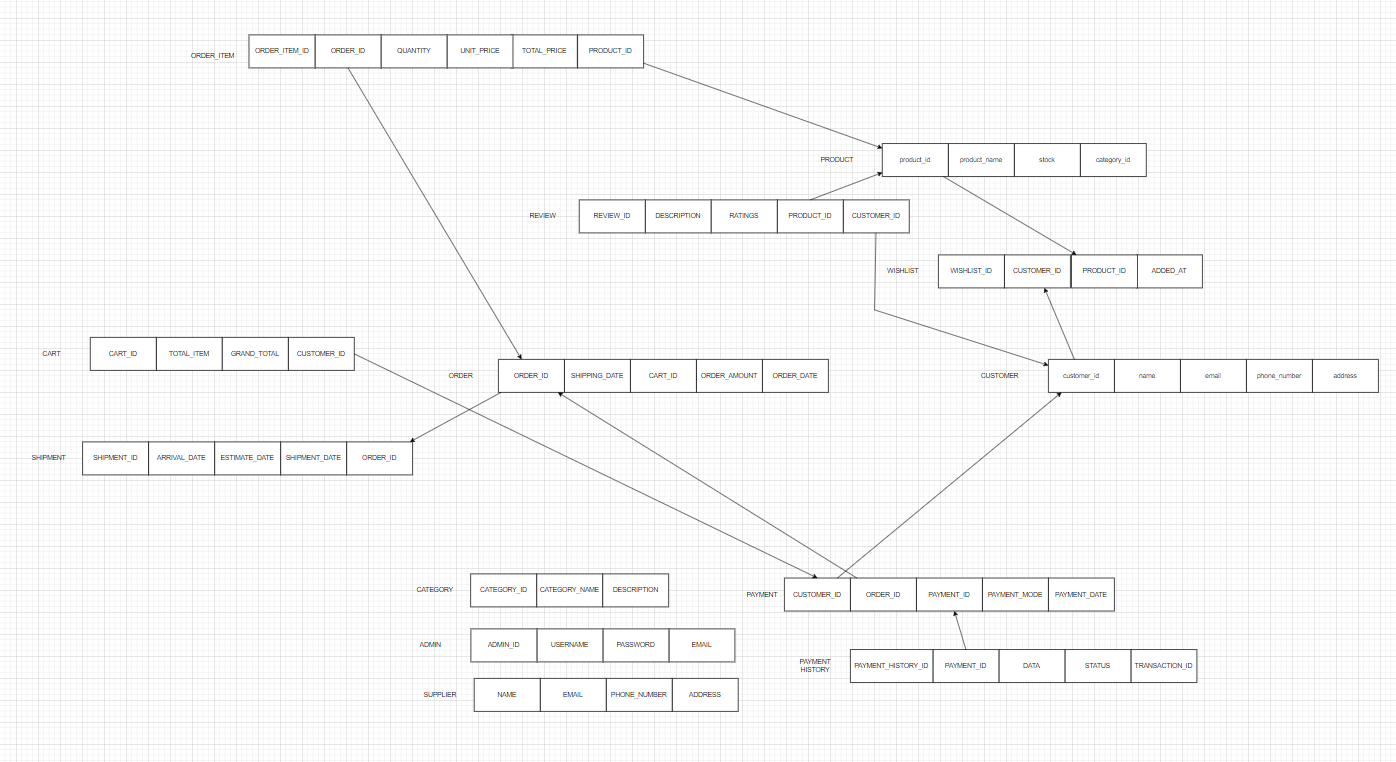
(5, 'Sports & Outdoors', 'Equipment and gear for sports, outdoor activities, and fitness.');

**III. Entity Relationship Diagram**

<https://drive.google.com/file/d/1KjTnjVbREhLznJChICxfK8Gu4VXYlaDH/view?usp=sharing>



**IV. Relational Model**



**V. Normalization**

DATABASE SCEMA-

* Customer (customer\_id, name, email, phone\_number, address\_id )
* Product (product\_id,product\_name, stock ,category\_id,MRP,Brand)
* Orders (order\_id,shipping\_date , cart\_id , order\_amount ,order\_date )
* OrderItem (order\_item\_id,order\_id , product\_id, quantity , unit\_price , total\_price )
* payment ( customer\_id,order\_id , payment\_id ,payment\_mode ,payment\_date )
* payment\_history (payment\_history\_id ,payment\_id ,data ,status , transaction\_id )
* Admins ( admin\_id, username, password, email)
* cart ( cart\_id ,total\_item ,grand\_total, customer\_id )
* Supplier ( name, email ,phone\_number, address)
* review (review\_id,description,ratings ,product\_id , customer\_id )
* wishlist (wishlist\_id, customer\_id,product\_id, added\_at)
* shipment ( shipment\_id ,arrival\_date ,estimate\_date ,shipment\_date , order\_id )
* category (category\_id, category\_name, description )

To normalize the given database to 3rd Normal Form 93NF), we need to eliminate any transitive dependencies and ensure that each non-prime attribute is fully functionally dependent on the primary key.

First, lets identify the functional dependencies-

1. Address Table (to store customer addresses)
2. Customers Table (with a foreign key referencing the Addresses table):
3. Product Table (unchanged):
4. Orders Table (unchanged):
5. OrderItem Table (unchanged):
6. PaymentHistory Table (with a foreign key referencing the Payment table):
7. Admins Table (unchanged):
8. Cart Table (with a foreign key referencing the Customers table):
9. Supplier Table (unchanged):
10. Review Table (with foreign keys referencing the Product and Customers tables):
11. Wishlist Table (with foreign keys referencing the Product and Customers tables):
12. Shipment Table (with a foreign key referencing the Orders table):
13. Category Table (unchanged):

Now, let's normalize the database:

**First Normal Form (1NF):**

- Ensure that each attribute contains atomic values.

- There are no multivalued attributes in the given schema, so it's already in 1NF.

**Second Normal Form (2NF):**

- Ensure that there are no partial dependencies.

All the tables seem to have single attribute primary keys, so there are no partial dependencies. Therefore, it's already in 2NF.

**Third Normal Form (3NF):**

- Ensure that there are no transitive dependencies.

Tables after Normalization-

1. CREATE TABLE Addresses (

address\_id INT PRIMARY KEY,

street VARCHAR(255),

city VARCHAR(100),

state VARCHAR(100),

zip\_code VARCHAR(20),

country VARCHAR(100)

);

1. CREATE TABLE Customers (

customer\_id INT PRIMARY KEY,

name VARCHAR(100),

email VARCHAR(255),

phone\_number VARCHAR(20),

address\_id INT,

FOREIGN KEY (address\_id) REFERENCES Addresses(address\_id)

);

1. CREATE TABLE Product (

product\_id INT PRIMARY KEY,

product\_name VARCHAR(255),

stock INT,

category\_id INT,

MRP DECIMAL(10, 2),

Brand VARCHAR(100)

);

1. CREATE TABLE Orders (

order\_id INT PRIMARY KEY,

shipping\_date DATE,

cart\_id INT,

order\_amount DECIMAL(10, 2),

order\_date DATE

);

1. CREATE TABLE OrderItem (

order\_item\_id INT PRIMARY KEY,

order\_id INT,

product\_id INT,

quantity INT,

unit\_price DECIMAL(10, 2),

total\_price DECIMAL(10, 2),

FOREIGN KEY (order\_id) REFERENCES Orders(order\_id),

FOREIGN KEY (product\_id) REFERENCES Product(product\_id)

);

1. Payment Table (with foreign keys referencing Customers and Orders tables):

CREATE TABLE Payment (

payment\_id INT PRIMARY KEY,

customer\_id INT,

order\_id INT,

payment\_mode VARCHAR(50),

payment\_date DATE,

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id),

FOREIGN KEY (order\_id) REFERENCES Orders(order\_id)

);

1. CREATE TABLE PaymentHistory (

payment\_history\_id INT PRIMARY KEY,

payment\_id INT,

data VARCHAR(255),

status VARCHAR(50),

transaction\_id VARCHAR(100),

FOREIGN KEY (payment\_id) REFERENCES Payment(payment\_id)

);

1. CREATE TABLE Admins (

admin\_id INT PRIMARY KEY,

username VARCHAR(50),

password VARCHAR(255),

email VARCHAR(255)

);

1. CREATE TABLE Cart (

cart\_id INT PRIMARY KEY,

total\_item INT,

grand\_total DECIMAL(10, 2),

customer\_id INT,

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)

);

1. CREATE TABLE Supplier (

supplier\_id INT PRIMARY KEY,

name VARCHAR(100),

email VARCHAR(255),

phone\_number VARCHAR(20),

address VARCHAR(255)

);

1. CREATE TABLE Review (

review\_id INT PRIMARY KEY,

description VARCHAR(255),

ratings DECIMAL(3, 2),

product\_id INT,

customer\_id INT,

FOREIGN KEY (product\_id) REFERENCES Product(product\_id),

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)

);

1. CREATE TABLE Wishlist (

wishlist\_id INT PRIMARY KEY,

customer\_id INT,

product\_id INT,

added\_at TIMESTAMP,

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id),

FOREIGN KEY (product\_id) REFERENCES Product(product\_id)

);

1. CREATE TABLE Shipment (

shipment\_id INT PRIMARY KEY,

arrival\_date DATE,

estimate\_date DATE,

shipment\_date DATE,

order\_id INT,

FOREIGN KEY (order\_id) REFERENCES Orders(order\_id)

);

1. CREATE TABLE Category (

category\_id INT PRIMARY KEY,

category\_name VARCHAR(50),

description VARCHAR(255)

);

**VI. SQL Queries**

Using a DBMS software (SQLite3 or MySQL or any other of your choice):

* Create the tables
* Populate the tables (insert some meaningful data, at least 10 tuples for each relation)
* Run SQL queries (minimum 20) covering **all concepts** learned in the class

This section should contain the question, SQL code, and the output snapshot for each query.

**VI. Project demonstration**

* Tools/software/ libraries used
* Screenshot and Description of the Demonstration of project ( If GUI is made)

**VII. Self -Learning beyond classroom**

:

We learned to design and implement a comprehensive database system for a grocery store, including tables for customers, products, orders, and payments. I also explored creating relationships between tables using foreign keys and learned how to insert data into these tables. This hands-on experience expanded my understanding of SQL and database design principles beyond the classroom.

**VIII. Learning from the Project**

This project significantly enhanced our understanding of database design, SQL, and the practical application of these skills in real-world scenarios. It provided me with hands-on experience in creating and managing complex databases, which is crucial for any data-driven project. Additionally, it improved my problem-solving skills, as I had to design the database to efficiently handle various operations such as inventory management, customer relationship management, and transaction processing. This project also introduced me to the importance of data integrity, security, and user-friendly interfaces in database applications. Overall, it has prepared me well for future projects and roles in the tech industry.

**IX. Challenges Faced**

The challenges faced during the development of this project included:

1. **Designing the Database Schema**: Creating a schema that accurately represented the relationships between different entities (e.g., customers, products, orders) while ensuring data integrity and efficiency was challenging.

2. **Managing Foreign Keys and Relationships**: Establishing and maintaining the correct relationships between tables, especially when dealing with foreign keys, was complex and required careful planning to avoid data inconsistencies.

3.**Data Normalization**: Ensuring that the database was normalized to reduce redundancy and improve data integrity was a challenge. It required a good understanding of normalization rules and the ability to identify and eliminate redundancies.

4. **Security and Data Protection**: Implementing security measures to protect sensitive data, such as customer information and payment details, was crucial. This included understanding and applying encryption techniques and access controls.

5. **Performance Optimization**: Ensuring that the database could handle the expected load and perform efficiently, especially with complex queries and large datasets, was a challenge. This required optimizing queries and indexing strategies.

6. **User Interface Design**: Designing a user-friendly interface that allowed users to easily interact with the database was challenging. It required a good understanding of user experience (UX) principles and the ability to create intuitive and responsive designs.

7. **Integration with Backend and Frontend**: Integrating the database with the backend and frontend components of the application was complex. It required understanding how to connect the database to the application logic and how to handle data transfer between the frontend and backend.

8. **Handling Errors and Exceptions**: Implementing robust error handling and exception management was challenging. It required anticipating potential issues and designing the system to handle them gracefully, ensuring a smooth user experience.

9. **Testing and Debugging**: Thoroughly testing the database and the application to identify and fix bugs was a significant challenge. It required a meticulous approach to ensure that all functionalities worked as expected and that the system was reliable and secure.

10. **Learning New Technologies**: The project required learning and integrating new technologies and tools, such as specific database management systems, programming languages, and frameworks. This was a steep learning curve, especially when dealing with complex features and functionalities.

**X. Conclusion**

1. **Database Design and Management**: Gained proficiency in designing and managing complex databases, including creating tables, relationships, and ensuring data integrity.

2. **SQL Proficiency**: Enhanced SQL skills, including writing efficient queries, managing data, and optimizing database performance.

3. **Software Development Practices**: Learned the importance of planning, designing, and testing software applications to ensure they meet user needs and are robust and secure.

4. **Problem-Solving and Critical Thinking**: Developed strong problem-solving and critical thinking skills, essential for tackling complex technical challenges and finding innovative solutions.

5. **Collaboration and Communication**: Improved collaboration and communication skills, crucial for working in team environments and effectively conveying technical concepts to non-technical stakeholders.