ONLINE SHOPPING MANAGEMENT SYSTEM

(PROJECT FILE)

MADE BY- Tanya

University- Delhi Technological University

DATABASE MANAGEMENT SYSTEM(DBMS)-

A Database Management System (DBMS) is a software system that enables the creation, management, and manipulation of databases. It provides a systematic and organized way to store, retrieve, and manage data, ensuring data integrity, security, and efficiency. Key functions of a DBMS include: Data Storage and Retrieval: Efficiently stores large amounts of data and provides mechanisms to quickly retrieve specific data when needed.

**Data Manipulation:** Allows users to insert, update, delete, and query data using languages such as SQL (Structured Query Language).

**Data Security:** Ensures that only authorized users can access and modify the data, protecting it from unauthorized access and breaches.

**Data Integrity:** Maintains data accuracy and consistency through constraints and rules, ensuring reliable data across the database.

**Transaction Management:** Supports concurrent access and ensures that transactions are processed reliably and adhere to the ACID properties (Atomicity, Consistency, Isolation, Durability).

BENEFITS OF DATABASE MANAGEMENT SYSTEM-

**Improved Data Sharing:** Multiple users can access the database simultaneously while maintaining data integrity.

**Better Data Integration:** Centralizes data, reducing redundancy and inconsistencies.

**Enhanced Data Security:** Implements robust security measures to protect sensitive data.

**Data Consistency:** Ensures that data remains consistent and accurate across different operations and transactions.

**Backup and Recovery:** Provides mechanisms to recover data in case of failures or disasters.

Popular DBMS examples include MySQL, PostgreSQL, Oracle, and Microsoft SQL Server. Each of these systems supports different types of databases and offers various features tailored to specific needs and applications. Overall, a DBMS is crucial for effective data management in organizations, enabling better decision-making and streamlined operations.

**SOUCE CODE** (ONLINE SHOPPING MANAGEMENT SYSTEM)-

-- Create the database

CREATE DATABASE IF NOT EXISTS Online Shopping;

USE Online Shopping;

-- Create Users table

CREATE TABLE IF NOT EXISTS Users (

UserID INT PRIMARY KEY AUTO\_INCREMENT,

Username VARCHAR(50) NOT NULL UNIQUE,

Password VARCHAR(255) NOT NULL,

Email VARCHAR(100) NOT NULL UNIQUE,

FirstName VARCHAR(50),

LastName VARCHAR(50),

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- Create Categories table

CREATE TABLE IF NOT EXISTS Categories (

CategoryID INT PRIMARY KEY AUTO\_INCREMENT,

CategoryName VARCHAR(100) NOT NULL

);

-- Create Products table

CREATE TABLE IF NOT EXISTS Products (

ProductID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(100) NOT NULL,

Description TEXT,

Price DECIMAL(10, 2) NOT NULL,

Stock INT NOT NULL,

CategoryID INT,

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID)

);

-- Create Orders table

CREATE TABLE IF NOT EXISTS Orders (

OrderID INT PRIMARY KEY AUTO\_INCREMENT,

UserID INT,

OrderDate TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

TotalAmount DECIMAL(10, 2) NOT NULL,

Status VARCHAR(50) DEFAULT 'Pending',

FOREIGN KEY (UserID) REFERENCES Users(UserID)

);

-- Create OrderDetails table

CREATE TABLE IF NOT EXISTS OrderDetails (

OrderDetailID INT PRIMARY KEY AUTO\_INCREMENT,

OrderID INT,

ProductID INT,

Quantity INT NOT NULL,

Price DECIMAL(10, 2) NOT NULL,

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),

FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

);

-- Create Payments table

CREATE TABLE IF NOT EXISTS Payments (

PaymentID INT PRIMARY KEY AUTO\_INCREMENT,

OrderID INT,

PaymentDate TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

Amount DECIMAL(10, 2) NOT NULL,

PaymentMethod VARCHAR(50),

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)

);

-- Create Reviews table

CREATE TABLE IF NOT EXISTS Reviews (

ReviewID INT PRIMARY KEY AUTO\_INCREMENT,

UserID INT,

ProductID INT,

Rating INT CHECK (Rating >= 1 AND Rating <= 5),

Comment TEXT,

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (UserID) REFERENCES Users(UserID),

FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

);

-- Create Cart table

CREATE TABLE IF NOT EXISTS Cart (

CartID INT PRIMARY KEY AUTO\_INCREMENT,

UserID INT,

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (UserID) REFERENCES Users(UserID)

);

-- Create CartItems table

CREATE TABLE IF NOT EXISTS CartItems (

CartItemID INT PRIMARY KEY AUTO\_INCREMENT,

CartID INT,

ProductID INT,

Quantity INT NOT NULL,

FOREIGN KEY (CartID) REFERENCES Cart(CartID),

FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

);

-- Create Addresses table

CREATE TABLE IF NOT EXISTS Addresses (

AddressID INT PRIMARY KEY AUTO\_INCREMENT,

UserID INT,

AddressLine1 VARCHAR(255) NOT NULL,

AddressLine2 VARCHAR(255),

City VARCHAR(100) NOT NULL,

State VARCHAR(100) NOT NULL,

ZipCode VARCHAR(20) NOT NULL,

Country VARCHAR(100) NOT NULL,

FOREIGN KEY (UserID) REFERENCES Users(UserID)

);

-- Create Shipping table

CREATE TABLE IF NOT EXISTS Shipping (

ShippingID INT PRIMARY KEY AUTO\_INCREMENT,

OrderID INT,

AddressID INT,

ShippingMethod VARCHAR(100),

ShippingDate TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

DeliveryDate TIMESTAMP,

Status VARCHAR(50) DEFAULT 'Pending',

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),

FOREIGN KEY (AddressID) REFERENCES Addresses(AddressID)

);

-- Example of inserting data

INSERT INTO Categories (CategoryName) VALUES

('Electronics'),

('Clothing'),

('Books');

INSERT INTO Products (Name, Description, Price, Stock, CategoryID) VALUES

('Laptop', 'A high performance laptop', 50000, 10, 1),

('Dress', 'A beautiful dress', 1999, 50, 2),

('Novel', 'A bestselling novel', 9.99, 100, 3);

-- Example of inserting users

INSERT INTO Users (Username, Password, Email, FirstName, LastName) VALUES

('Kevin\_Taylor', 'password123', 'kevin@example.com', 'Kevin', 'Taylor'),

('Dev\_Ahuja', 'passvword456', 'dev@example.com', 'Dev', 'Ahuja');

-- Example of creating an order

INSERT INTO Orders (UserID, TotalAmount, Status) VALUES

(1, 50000, 'Pending');

-- Example of adding order details

INSERT INTO OrderDetails (OrderID, ProductID, Quantity, Price) VALUES

(1, 1, 1, 1999),

(1, 3, 2, 9.99);

-- Example of adding a payment

INSERT INTO Payments (OrderID, Amount, PaymentMethod) VALUES

(1, 50000, 'Credit Card');

-- Example of adding a review

INSERT INTO Reviews (UserID, ProductID, Rating, Comment) VALUES

(1, 1, 5, 'Excellent laptop!');

-- Example of selecting data

SELECT \* FROM Products WHERE CategoryID = 1;

-- Example of viewing orders

SELECT o.OrderID, u.Username, o.TotalAmount, o.Status

FROM Orders o

JOIN Users u ON o.UserID = u.UserID;

**ER DIAGRAM –**

An Entity-Relationship (ER) diagram is a visual representation of the data and relationships within a database system. It is a key tool used in database design to illustrate the logical structure of the database, making it easier to understand and communicate the data requirements and relationships among various entities.

Key Components of an ER Diagram:

**Entities:**

Represent objects or concepts that have distinct and independent existence in the database. Entities are usually depicted as rectangles and named using singular nouns (e.g., Customer, Order).

**Attributes:**

Describe properties or characteristics of entities. Attributes are often shown as ovals connected to their respective entities by lines (e.g., Customer Name, Order Date).

**Relationships:**

Represent associations between entities. Relationships are typically illustrated as diamonds or by connecting lines between entities, and they often include verbs to describe the nature of the relationship (e.g., Places, Contains).

**Primary Keys:**

Unique identifiers for entities, ensuring each instance of an entity is distinguishable. Primary keys are often underlined within the entity rectangle.

**Cardinality:**

Indicates the number of instances of one entity that can be associated with instances of another entity.

**ER DIAGRAM (ONLINE SHOPPING MANAGEMENT SYSTEM)-**

