

# CampusCart

## PROJECT REPORT

---

- JASON DMELLO  
(210905181)
- KRIISH SOLANKI  
(210905158)

# CONTENTS

---

CONTENTS .....	2
ABSTRACT .....	3
PROBLEM STATEMENT .....	4
ER DIAGRAM .....	5
DDL COMANDS TO CREATE TABLE .....	9
SQL QUERIES.....	10
UI DESIGN .....	12
DATABASE CONNECTIVITY .....	15
PLSQL/TRIGGERS.....	17
REFERENCES .....	18

# ABSTRACT

---

CampusCart is an e-commerce web application designed for a university campus community. The application allows users to purchase various products. The goal of CampusCart is to provide a convenient and efficient platform for campus community members to purchase items they need on campus without the need to leave the campus.

The application includes features such as user registration, product browsing, order history tracking, and product rating and review.

The project was developed using a MySQL database, Python Flask framework for the backend, and HTML, CSS, and JavaScript for the frontend. The application also includes the use of PL/SQL to handle database operations and ensure data integrity.

Overall, CampusCart provides a reliable and efficient platform for campus community members to purchase products and services on campus, contributing to a more convenient and productive campus experience.

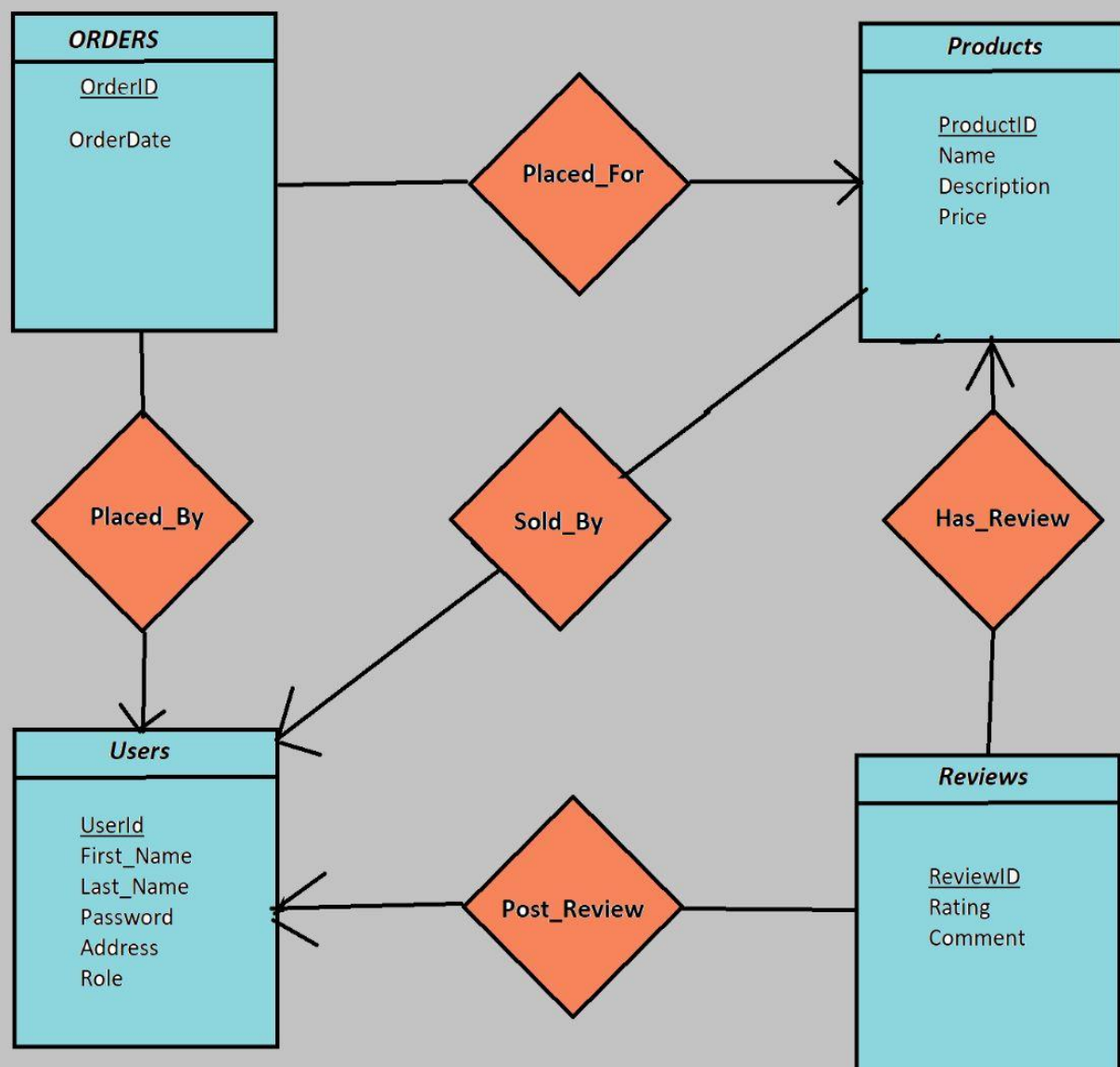
# PROBLEM STATEMENT

---

The objective of this project is to develop a robust and efficient database management system for the CampusCart e-commerce web application.

# ER DIAGRAM

---



# RELATIONAL TABLES & SAMPLE DATA

---

- Table: Users

UserID	FirstName	LastName	Email	Password	Address	Role
1	Kriish	Solanki	k_g_solanki@yahoo.in	abc123	Block 18	buyer
2	Jason	Dmello	jason.dmello21@gmail.com	123abc	Block 18	seller
3	Abrham	Silbrachatz	abraham.silbraschatz@db.com	mypassw0rd	AB5	seller

- Table: Products

ProductID	Name	Description	Price	SellerID
1	Fundamental of Database Systems	Sixth Edition contain the fundamentals of dbms that help students to acquire cast knowledge of this topic. From beginner to professional	799.00	2
2	R.D Sharma Maths	R.D Sharma is very famous among IIT-JEE aspirants.Every student refers it for 10+2 level exams like: · IIT-JEE · AIEEE · Medical · Olympiad and other exams.	699.99	3
3	Database System Concepts	Used by colleges like MIT Manipal to teach students	249.00	2

- Table: Orders

OrderID	ProductID	OrderDate	UserID
1	2	2022-01-01 10:30:00	1
2	1	2022-01-02 11:45:00	2
3	3	2022-01-03 12:15:00	1

- Table: Reviews

ReviewID	UserID	ProductID	Rating	Comment
1	1	2	4	"Great Book! Easy to understand and amazing quality."
2	2	1	5	"Love the new edition."
3	3	3	3	"Good read, but not worth the price."



# DDL COMANDS TO CREATE TABLE

---

```
CREATE TABLE Users (  
    UserID INT PRIMARY KEY AUTO_INCREMENT,  
    FirstName VARCHAR(50) NOT NULL,  
    LastName VARCHAR(50) NOT NULL,  
    Email VARCHAR(50) UNIQUE NOT NULL,  
    Password VARCHAR(100) NOT NULL,  
    Address VARCHAR(100) NOT NULL,  
    Role ENUM('buyer', 'seller') NOT NULL  
);  
  
CREATE TABLE Products (  
    ProductID INT PRIMARY KEY AUTO_INCREMENT,  
    Name VARCHAR(100) NOT NULL,  
    Description VARCHAR(255) NOT NULL,  
    Price DECIMAL(10,2) NOT NULL,  
    SellerID INT NOT NULL,  
    FOREIGN KEY (SellerID) REFERENCES Users(UserID)  
);  
  
CREATE TABLE Orders (  
    OrderID INT PRIMARY KEY AUTO_INCREMENT,  
    ProductID INT,  
    OrderDate DATETIME NOT NULL DEFAULT NOW(),  
    UserID INT NOT NULL,  
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID),  
    FOREIGN KEY (UserID) REFERENCES Users(UserID)  
);  
  
CREATE TABLE Reviews (  
    ReviewID INT PRIMARY KEY AUTO_INCREMENT,  
    UserID INT NOT NULL,  
    ProductID INT NOT NULL,  
    Rating INT NOT NULL,  
    Comment TEXT,  
    FOREIGN KEY (UserID) REFERENCES Users(UserID),  
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)  
);  
,
```

# SQL QUERIES

---

```
--Allow users to create a profile
INSERT INTO Users(UserID, FirstName, LastName, Email, Address) VALUES (%s, %s, %s, %s, %s);

--Allow users to view their personal information.
SELECT * FROM Users WHERE UserID = %s;

--Allow users to log in
SELECT * FROM Users WHERE Email = %s AND Password = %s;

--Search functionality - Allows users to search for products by name
SELECT * FROM Products WHERE Name LIKE '%search_term%';

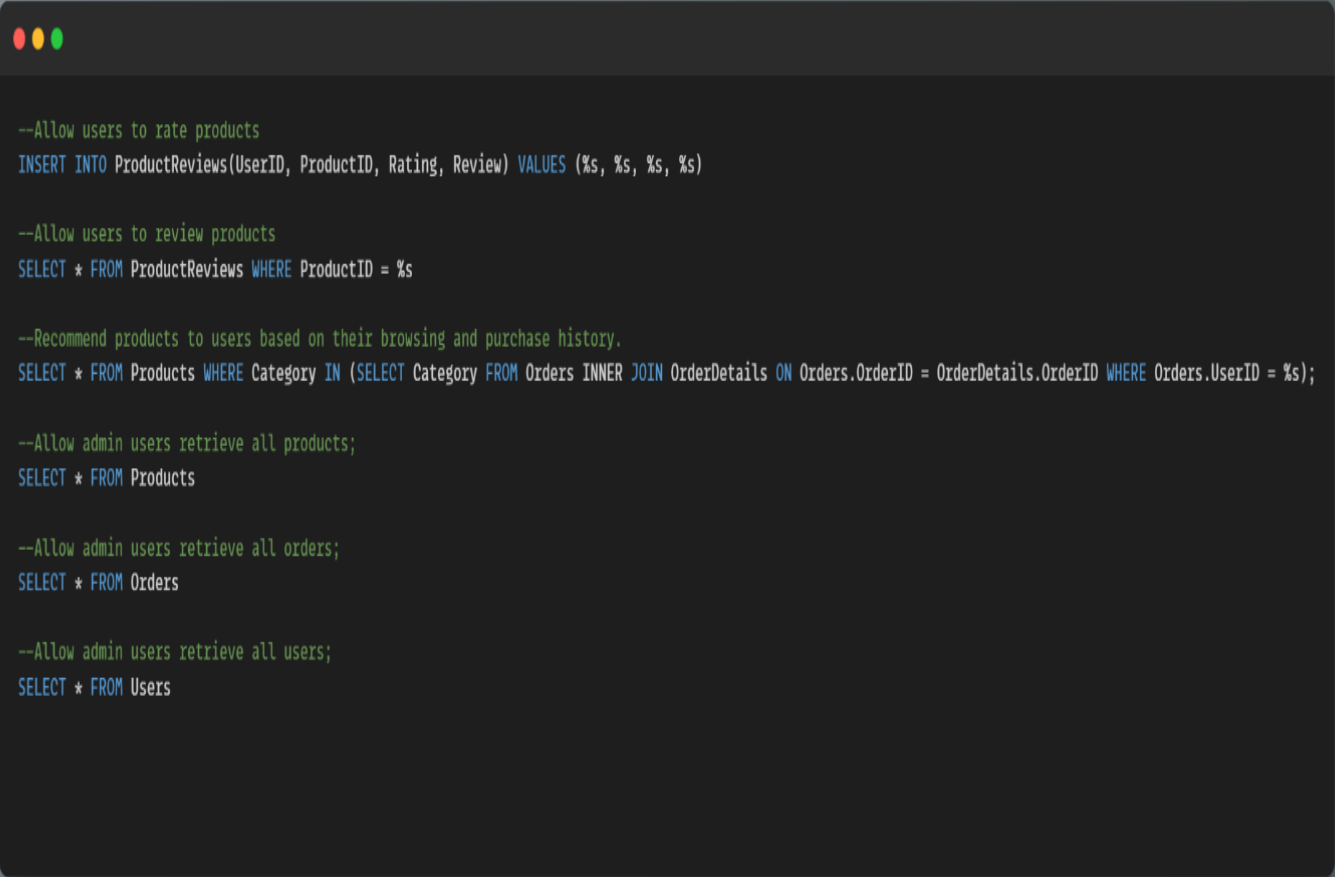
--Search functionality - Allows users to search for products by category
SELECT * FROM Products WHERE Category LIKE '%search_term%';

--Sorting functionality - Allows users to sort products by price
SELECT * FROM Products ORDER BY Price DESC;

--Sorting functionality - Allows users to sort products by rating
SELECT * FROM Products ORDER BY Rating DESC; -- sort products by descending rating

--Order history functionality - Allows users to view their previous orders.
SELECT * FROM Orders WHERE UserID = %s; -- get all orders made by user with ID 1

SELECT * FROM Orders o
JOIN OrderItems oi ON o.OrderID = oi.OrderID
JOIN Products p ON oi.ProductID = p.ProductID
WHERE o.UserID = 1; -- get all products in all orders made by user with ID 1
```



```
--Allow users to rate products
INSERT INTO ProductReviews(UserID, ProductID, Rating, Review) VALUES (%s, %s, %s, %s)

--Allow users to review products
SELECT * FROM ProductReviews WHERE ProductID = %s

--Recommend products to users based on their browsing and purchase history.
SELECT * FROM Products WHERE Category IN (SELECT Category FROM Orders INNER JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID WHERE Orders.UserID = %s);

--Allow admin users retrieve all products;
SELECT * FROM Products

--Allow admin users retrieve all orders;
SELECT * FROM Orders

--Allow admin users retrieve all users;
SELECT * FROM Users
```

# UI DESIGN

---

## Sign Up



Create Account

Already have an account? [Log In](#)

# Login

Email

abraham.silbraschatz@db.com

Password

•••••

Login

Don't have an account? [Sign Up](#)

# What will you be buying today?

Database

## Database System Concepts

Used by colleges like MIT  
Manipal to teach  
students

**999.00**

## Fundamental of Database Systems

Sixth Edition contain the  
fundaments of dbms that  
help students to acquire  
cast knowledge of this  
topic. From beginner to  
professional.

**899.00**

## Profile

### Order History

Product Name	Date Ordered	Rating
Windows 11	Fri, 12 May 2023 00:00:00 GMT	<div>--v</div> <div>Submit Review</div>

### Add Item

**Product Name**

**Description**

**Price**

Add Item

### See Reviews

Product Name	Comment	Rating
Database System Concepts	Used this book for my studies. Could use a bit more explanation on its part	3
Fundamental of Database Systems	My goto book for studying dbms	5
Database System Concepts	An amazing book. With the right mentor, one can become a master of dbms	5

# DATABASE CONNECTIVITY

---

```
-- Connecting backend to the database

app = Flask(_name_)
app.secret_key = "Sec0"
CORS(app, resources={r"/*": {"origins": "http://localhost:3000"}})

mydb = mysql.connector.connect(
    host="127.0.0.1", user="root", password="", database="campuscart"
)
```

```
--To send request to the backend
const handleSubmit = async event => {
  event.preventDefault();
  await axios.post('http://127.0.0.1:5000/signup', { first_name, last_name, email, password, address, type })
    .then(response => {
      setMessage(response.data['message']);
      console.log(message)
    })
    .catch(error => {
      setError(error);
      console.log(error)
    });
}
```

```

--Gets request from the frontend to send data into the users table
@app.route("/signup", methods=["GET", "POST", "OPTIONS"])
def signup():
    if request.method == "POST":
        # Add your SQL query to insert the student's data into the database
        json_data = json.loads(request.data.decode('utf-8'))
        mycursor = mydb.cursor()
        sql = "INSERT INTO Users(FirstName, LastName, Email, Password, Address, Role) VALUES (%s, %s, %s, %s, %s, %s)"
        val = (
            json_data["first_name"],
            json_data["last_name"],
            json_data["email"],
            json_data["password"],
            json_data["address"],
            json_data["type"]
        )
        mycursor.execute(sql, val)
        mydb.commit()

        response = jsonify({"message": "Successfully signed up"})
        return response
    response = jsonify({"message": "Cant perform this action"})
    return response

```



# PLSQL/TRIGGERS

---

```
--Table creation for average rating
CREATE TABLE AverageRatings (
    ProductID INT PRIMARY KEY,
    AvgRating DECIMAL(3,2),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);

--Updates rating
CREATE TRIGGER `update_average_rating` AFTER INSERT ON `reviews`
FOR EACH ROW
UPDATE averageratings
    SET AvgRating = (SELECT AVG(Rating) FROM Reviews WHERE ProductID = NEW.ProductID)
    WHERE ProductID = NEW.ProductID
I

--Inserts new product into AverageRatings
CREATE TRIGGER `insert_product_trigger` AFTER INSERT ON `products`
FOR EACH ROW
insert INTO averageratings(ProductID, AvgRating) VALUES (NEW.ProductID, 0)
```

# REFERENCES

---

- [ReactJS documentation](#)
- [XAMPP documentation](#)
- [MYSQL documentation](#)
- [Flask documentation](#)
- [NodeJS documentation](#)
- [Fundamentals of Database Systems](#)