Project name:

Plant Nursery Management System.

Project group name:

Ghorsa project.

Group name:

Tamara Tumeh(Shiren Hijazi).

Tojan Naiem(Zaina AlSaed).

Ruba Alhaj(Shiren Hijazi).

- **Project Description:**

Ghorsa is an e-commerce platform specializing in selling plants and gardening products. It offers users an organized shopping experience, starting with a categorized catalog of plants and extending to a seamless checkout process. Users can register, manage multiple delivery addresses, and browse products with details like name, price, description, and stock availability. The platform allows users to add items to their shopping cart, mark products as favorites, and place orders that track their status from "Pending" to "Completed." Payments are securely processed and linked to each order. Users can add multiple addresses to their profiles for flexible delivery options. The website showcases trending products and the latest arrivals, providing a dynamic shopping experience. In addition to plants, Ghorsa also offers a variety of gardening tools and supplies, ensuring users have access to everything they need for their gardening endeavors.

Category Table

- Stores information about product categories.
- **Fields**:
    - category_id: Primary key, unique identifier for each category.
    - name: Name of the category.
    - image: Image associated with the category.

Product Table

- Represents the products available in the system.
- **Fields**:
    - product_id: Primary key, unique identifier for each product.
    - name: Name of the product.
    - price: Price of the product.

    - description: Description of the product.
    - pot_color: Color of the pot for plant-related products.
    - plant_care: Plant care instructions.
    - stock: Quantity available in inventory.
    - category_id: Foreign key linking to the category table.

User Table

- Represents users of the system.
- **Fields**:
    - user_id: Primary key, unique identifier for each user.
    - First_Name: First name of the user.
    - Last_Name: Last name of the user.
    - email: Email address of the user.
    - password: User's password for authentication.
    - Phone: Contact phone number.

Address Table

- Stores addresses associated with users.
- **Fields**:
    - address_id: Primary key, unique identifier for each address.
    - city: City of the address.
    - country: Country of the address.
    - zipCode: Postal code of the address.
    - user_id: Foreign key linking to the user table.

Order Table

- Tracks orders placed by users.

- **Fields**:
    - order_id: Primary key, unique identifier for each order.
    - OrderDate: Date the order was placed.
    - status: Current status of the order (e.g., pending, shipped).
    - order_amount: Total amount for the order.
    - user_id: Foreign key linking to the user table.

Payment Table

- Manages payment details for orders.
- **Fields**:
    - payment_id: Primary key, unique identifier for each payment.
    - amount: Payment amount.
    - payment_method: Method of payment (e.g., credit card, PayPal).
    - order_id: Foreign key linking to the order table.

Cart Table

- Manages shopping carts for users.
- **Fields**:
    - cart_id: Primary key, unique identifier for each cart.
    - tracking_id: Tracking information for the cart.
    - user_id: Foreign key linking to the user table.

User_Favorites Table

- Allows users to save their favorite items.
- **Fields**:
    - favorite_id: Primary key, unique identifier for each favorite item.
    - created_at: Timestamp when the favorite item was added.

- user_id: Foreign key linking to the user table.

Role Table

- Defines roles assigned to users.
- **Fields**:
  - id: Primary key, unique identifier for each role.
  - role_name: Name of the role (e.g., admin, customer).
  - user_id: Foreign key linking to the user table.

Each table is designed to efficiently manage specific data and is linked to others using foreign key relationships, ensuring data consistency and normalization.

- **Project scenario:**

A user registers on the system by providing their First_Name, Last_Name, email, password, and Phone number, which are stored in the user table under a unique user_id. The user adds multiple addresses, such as their home and office addresses, with details like city, country, and pin_Code, stored in the address table under a unique address_id.

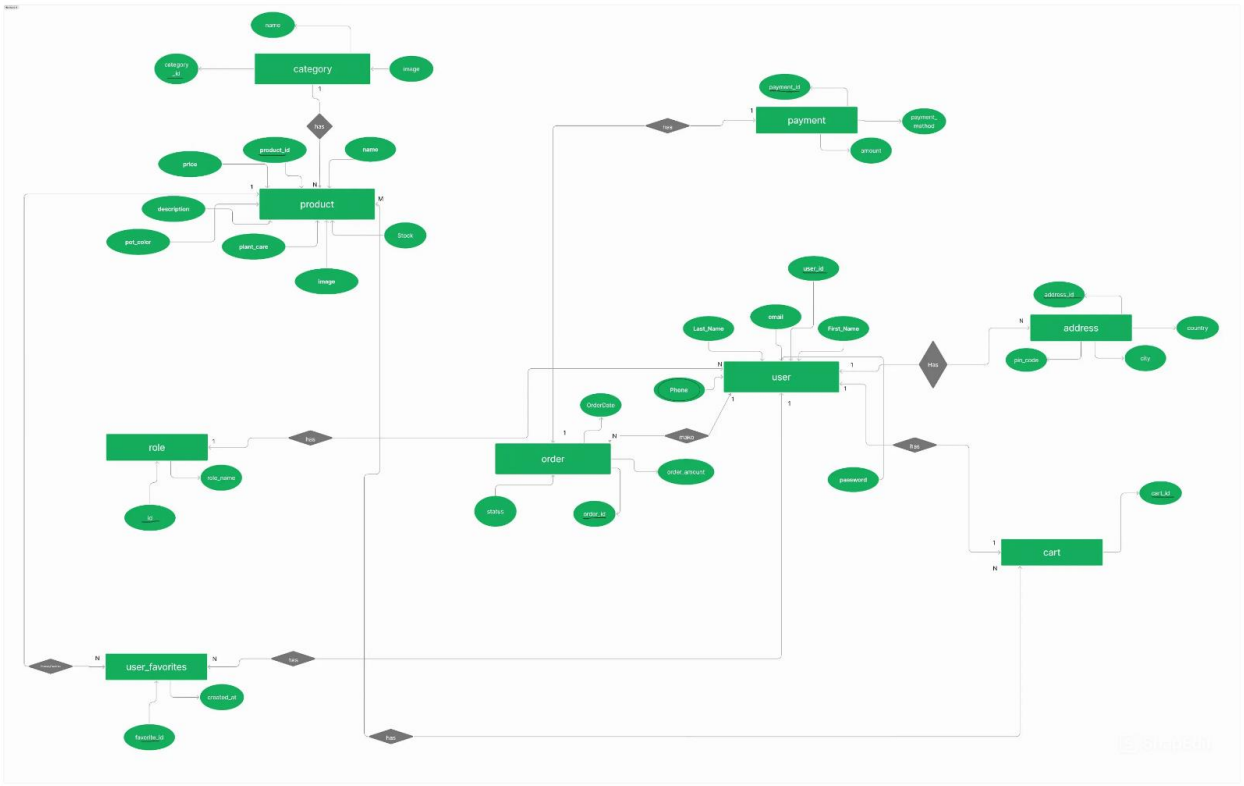
The user browses products in various categories such as "Indoor Plants" or "Outdoor Plants", managed in the category table.category are stored in the category table with attributes like name, image. Each category is identified by a category_id, and its products are stored in the product table with attributes like product_id, name, price, description, pot_color, plant_care,image, and stock.

The user adds products to their cart, represented in the cart table with a unique cart_id. The cart may contain multiple products, linked to the user table via user_id for tracking. Once the user places an order, the details are stored in the order table with a unique order_id, including the OrderDate, status, and order_amount. The order links to the user and their cart.

The user makes a payment, and the details, such as payment_id, amount, and payment_method, are stored in the payment table, linked to the order table. Additionally, the user can save favorite products in the user_favorites table , with attributes like created_at, and a unique favorite_id, enabling them to quickly revisit their preferred items.

Finally, the role table manages user roles, assigning roles such as "Admin" or "Customer" to users using role_name and a foreign key user_id. This structure ensures proper role-based access control within the system.

- **"Entity-Relationship Diagram" (ERD)**



Project code

CREATE DATABASE ghorsa;

USE ghorsa;

CREATE TABLE category(

category_id INT PRIMARY KEY AUTO_INCREMENT,

image VARCHAR(100) NOT NULL,

```
    name VARCHAR(30) NOT NULL
);

CREATE TABLE product(
    product_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(50) NOT NULL,
    description VARCHAR(500),
    plant_care VARCHAR(300),
    image VARCHAR(100) NOT NULL,
    pot_color VARCHAR(50),
    price DECIMAL(10,2) NOT NULL,
    stock INT DEFAULT 0 CHECK(stock >= 0),
    category_id INT,
    FOREIGN KEY (category_id) REFERENCES category(category_id)
);

CREATE TABLE roles (
    id INT PRIMARY KEY AUTO_INCREMENT,
    role_name VARCHAR(50) NOT NULL UNIQUE
);

CREATE TABLE user (
    user_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    phone varchar(15),
```

```
    role_id INT,
    FOREIGN KEY (role_id) REFERENCES roles(id)
);

CREATE TABLE address(
    address_id INT PRIMARY KEY AUTO_INCREMENT,
    country VARCHAR(20) NOT NULL,
    city VARCHAR(20) NOT NULL,
    street VARCHAR(100),
    pin_code INT NOT NULL,
    user_id INT,
    FOREIGN KEY (user_id) REFERENCES user(user_id)
);

CREATE TABLE order_table(
    order_id INT PRIMARY KEY AUTO_INCREMENT,
    order_amount DECIMAL(10,2) NOT NULL,
    order_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    status VARCHAR(20) DEFAULT 'Pending' CHECK(status IN ('Pending', 'Completed', 'Shipped')),
    user_id INT,
    FOREIGN KEY (user_id) REFERENCES user(user_id)
);

CREATE TABLE payment(
    payment_id INT PRIMARY KEY AUTO_INCREMENT,
    payment_method VARCHAR(50) DEFAULT 'Cash on Delivery',
    amount DECIMAL(10,2) NOT NULL,
```

```
    order_id INT,
    FOREIGN KEY (order_id) REFERENCES order_table(order_id)
);


CREATE TABLE cart(
    cart_id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT,
    FOREIGN KEY (user_id) REFERENCES user(user_id)
);

CREATE TABLE cart_item(
    cart_item_id INT PRIMARY KEY AUTO_INCREMENT,
    cart_id INT,
    price DECIMAL(10,2),
    product_id INT,
    quantity INT NOT NULL DEFAULT 0 CHECK(quantity >= 0),
    FOREIGN KEY (cart_id) REFERENCES cart(cart_id),
    FOREIGN KEY (product_id) REFERENCES product(product_id)
);

CREATE TABLE user_favorites (
    favorite_id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT NOT NULL,
    product_id INT NOT NULL,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES user(user_id),
```

```
    FOREIGN KEY (product_id) REFERENCES product(product_id),
    UNIQUE (user_id, product_id)
);
```