

# ONLINE SHOPPING APPLICATION

## SOURCE CODE:-

```
import java.util.*;

enum Category {
    ELECTRONICS,
    CLOTHING,
    BOOKS,
}

class Product {
    private String name;
    private double price;
    private Category category;

    public Product(String name, double price, Category
category) {
        this.name = name;
        this.price = price;
        this.category = category;
    }

    public String getName() {
        return name;
    }

    public double getPrice() {
        return price;
    }

    public Category getCategory() {
        return category;
    }
}

//Class representing a user
class User {
    private String username;
```

```
private String email;
private ShoppingCart cart;

public User(String username, String email) {
    this.username = username;
    this.email = email;
    this.cart = new ShoppingCart();
}

public String getUsername() {
    return username;
}

public String getEmail() {
    return email;
}

public ShoppingCart getCart() {
    return cart;
}
}

//Class representing a shopping cart
class ShoppingCart {
    private List<Product> products;

    public ShoppingCart() {
        products = new ArrayList<>();
    }

    public void addProduct(Product product) {
        products.add(product);
    }

    public void removeProduct(Product product) {
        products.remove(product);
    }

    public List<Product> getProducts() {
        return products;
    }

    public double calculateTotal() {
```

```

        double total = 0;
        for (Product product : products) {
            total += product.getPrice();
        }
        return total;
    }
}

//Class representing an online shopping system
class OnlineShopping {
    private List<Product> availableProducts;
    private List<User> users;

    public OnlineShopping() {
        availableProducts = new ArrayList<>();
        users = new ArrayList<>();
    }

    public void addProduct(Product product) {
        availableProducts.add(product);
    }

    public void removeProduct(Product product) {
        availableProducts.remove(product);
    }

    public List<Product> getAvailableProducts() {
        return availableProducts;
    }

    public void registerUser(User user) {
        users.add(user);
    }

    public List<User> getUsers() {
        return users;
    }
}

public class OnlineShipping {
    public static void main(String[] args) {
        OnlineShopping onlineShopping = new
OnlineShopping();
        Scanner scanner = new Scanner(System.in);
    }
}

```

```

        // Adding products to the online shopping
system
        onlineShopping.addProduct(new
Product("Laptop", 999.99, Category.ELECTRONICS));
        onlineShopping.addProduct(new Product("T-
Shirt", 19.99, Category.CLOTHING));
        onlineShopping.addProduct(new Product("Java
Book", 29.99, Category.BOOKS));

        // Register a user with input from the user
System.out.println("Enter username:");
String username = scanner.nextLine();
System.out.println("Enter email:");
String email = scanner.nextLine();

        User user1 = new User(username, email);
        onlineShopping.registerUser(user1);

        // Adding products to the user's shopping
cart with user input
        System.out.println("Available products:");
        List<Product> availableProducts =
onlineShopping.getAvailableProducts();
        for (int i = 0; i <
availableProducts.size(); i++) {
            System.out.println(i + ". " +
availableProducts.get(i).getName());
        }

        ShoppingCart cart = user1.getCart();
        boolean addMore = true;
        while (addMore) {
            System.out.println("Enter the number of
the product to add to the cart (or -1 to stop:");
            int productNumber = scanner.nextInt();
            scanner.nextLine(); // Consume the
newline character

            if (productNumber == -1) {
                addMore = false;
            } else if (productNumber >= 0 &&
productNumber < availableProducts.size()) {

```

```

cart.addProduct(availableProducts.get(productNumber));
        } else {
            System.out.println("Invalid product
number. Please enter a valid number.");
        }
    }

    // Displaying the products in the user's
    cart and calculating the total
    System.out.println("User: " +
user1.getUsername());
    System.out.println("Cart items:");
    for (Product product :
user1.getCart().getProducts()) {
        System.out.println("- " +
product.getName() + " - $" + product.getPrice());
    }
    System.out.println("Total: $" +
user1.getCart().calculateTotal());

    scanner.close(); // Close the scanner
}
}

```

## OUTPUT:-

```

Enter username:
amit kumar
Enter email:
abc@gmail.com
Available products:
0. Laptop
1. T-Shirt
2. Java Book
Enter the number of the product to add to the cart (or -
1 to stop):
0
Enter the number of the product to add to the cart (or -
1 to stop):

```

1

Enter the number of the product to add to the cart (or - 1 to stop):

2

Enter the number of the product to add to the cart (or - 1 to stop):

1

Enter the number of the product to add to the cart (or - 1 to stop):

-1

User: amit kumar

Cart items:

- Laptop - \$999.99
- T-Shirt - \$19.99
- Java Book - \$29.99
- T-Shirt - \$19.99

Total: \$1069.96