ECaseMerce

Tung Tran, Hieu Nguyen, Hoang Do, Lam Nguyen

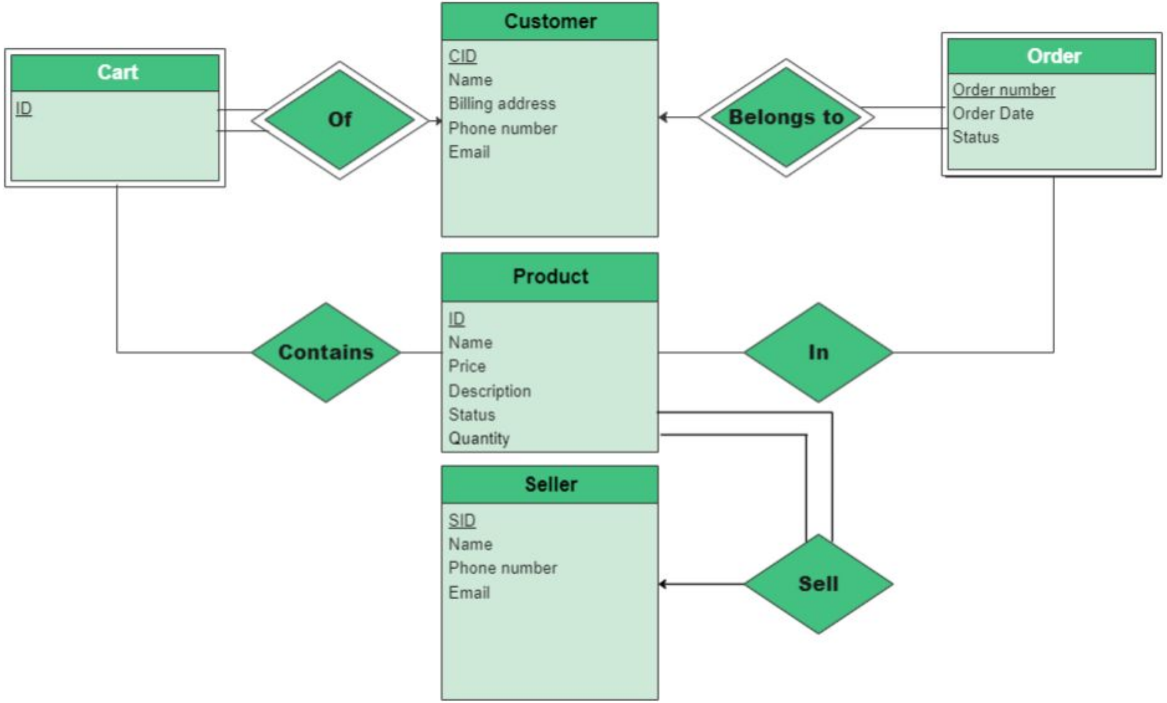
A CWRU student-to-student E-commerce website that offers:

- Authentication with CWRU's email
- Home page with a list of products that are being sold
- Search engine to search for on sale products
- Uploading your own products to sell
- Orders history, order status
- Shopping cart

Tech stack

- Front end: React.js, Redux
- Backend: Node.js, Express.js
- DBMS: MySQL

ER Diagram



Cart

ID

Of

Customer

CID
Name
Billing address
Phone number
Email

Belongs to

Order

Order number
Order Date
Status

Contains

Product

ID
Name
Price
Description
Status
Quantity

In

Seller

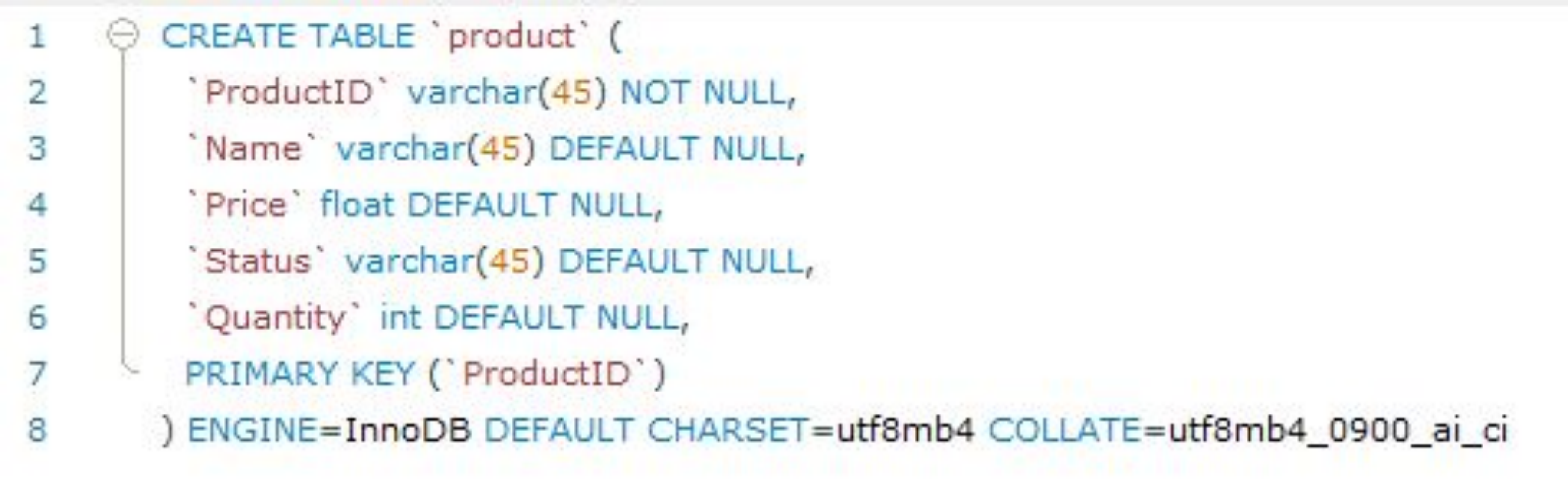
SID
Name
Phone number
Email

Sell

Data description

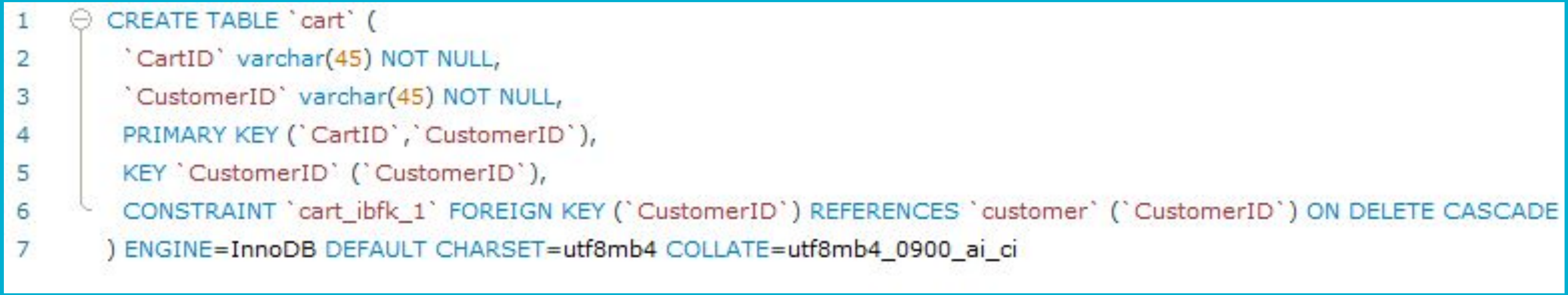
- We artificially generate our data based on the dataset about e-commerce downloaded from Kaggle.
- The dataset contains mostly all the fields we need, such as product name, product id, customer id, invoice number, invoice date.
- Use name generator tools to generate customer and seller name.
- Assign products to seller.
- https://www.kaggle.com/carrie1/ecommerce-data

Product



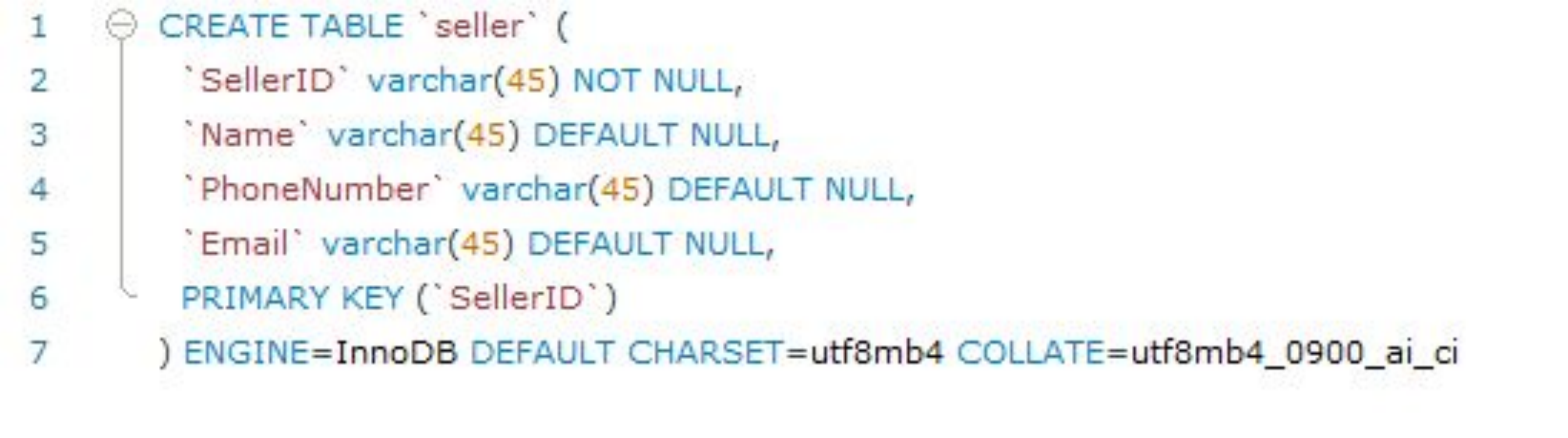
```sql
CREATE TABLE `product` (
  `ProductID` varchar(45) NOT NULL,
  `Name` varchar(45) DEFAULT NULL,
  `Price` float DEFAULT NULL,
  `Status` varchar(45) DEFAULT NULL,
  `Quantity` int DEFAULT NULL,
  PRIMARY KEY (`ProductID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Cart



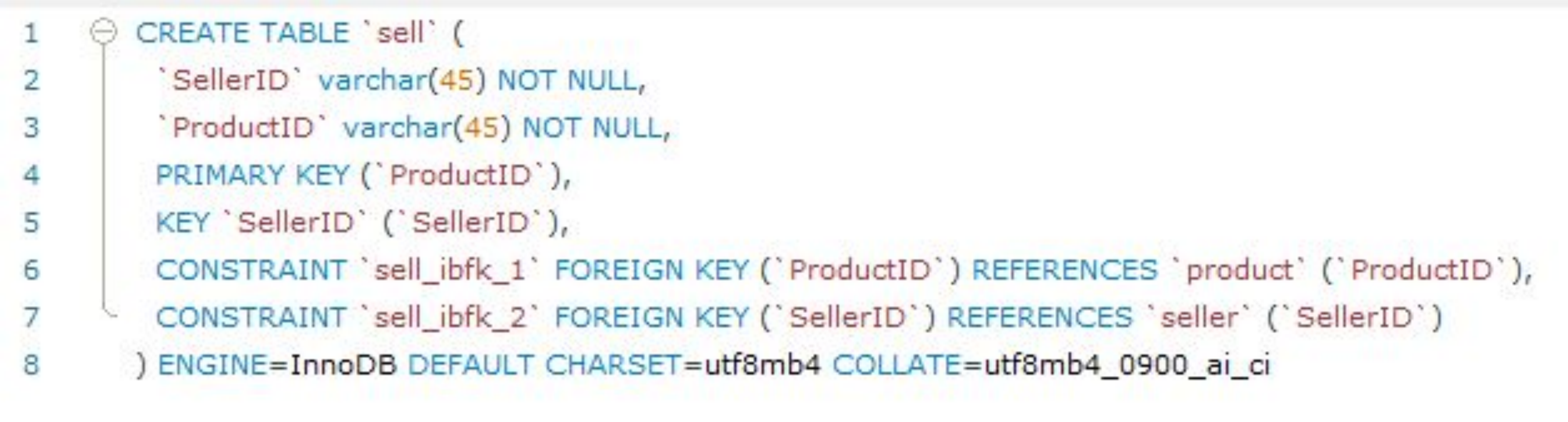
```
CREATE TABLE `cart` (
  `CartID` varchar(45) NOT NULL,
  `CustomerID` varchar(45) NOT NULL,
  PRIMARY KEY (`CartID`,`CustomerID`),
  KEY `CustomerID` (`CustomerID`),
  CONSTRAINT `cart_ibfk_1` FOREIGN KEY (`CustomerID`) REFERENCES `customer` (`CustomerID`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Seller



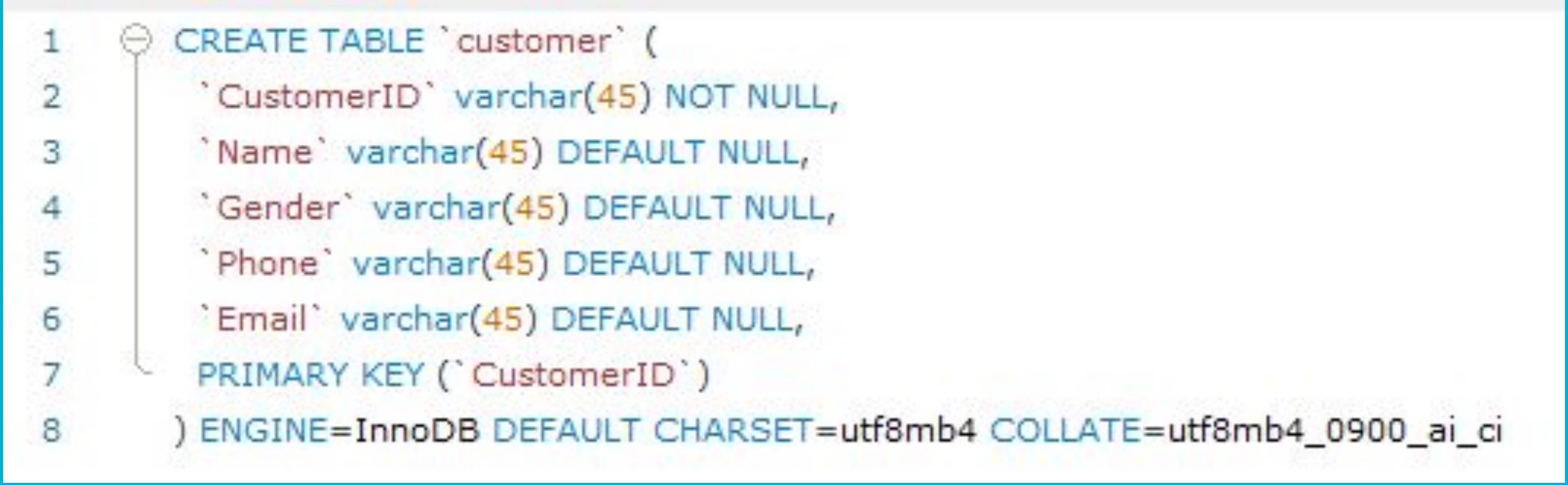
```
CREATE TABLE `seller` (
  `SellerID` varchar(45) NOT NULL,
  `Name` varchar(45) DEFAULT NULL,
  `PhoneNumber` varchar(45) DEFAULT NULL,
  `Email` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`SellerID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Sell



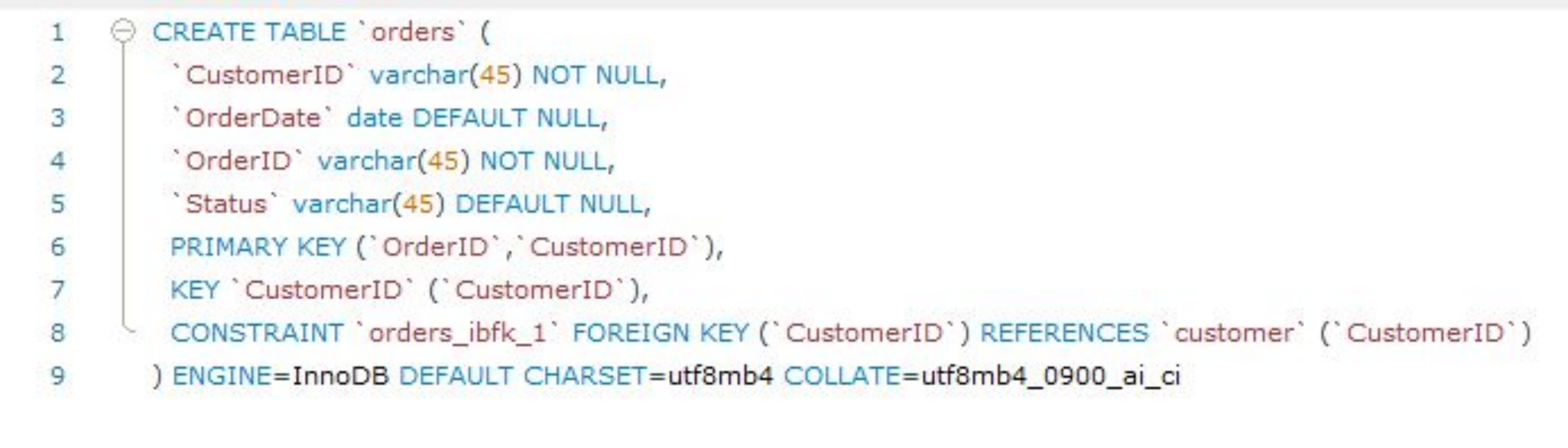
```sql
CREATE TABLE `sell` (
  `SellerID` varchar(45) NOT NULL,
  `ProductID` varchar(45) NOT NULL,
  PRIMARY KEY (`ProductID`),
  KEY `SellerID` (`SellerID`),
  CONSTRAINT `sell_ibfk_1` FOREIGN KEY (`ProductID`) REFERENCES `product` (`ProductID`),
  CONSTRAINT `sell_ibfk_2` FOREIGN KEY (`SellerID`) REFERENCES `seller` (`SellerID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Customer



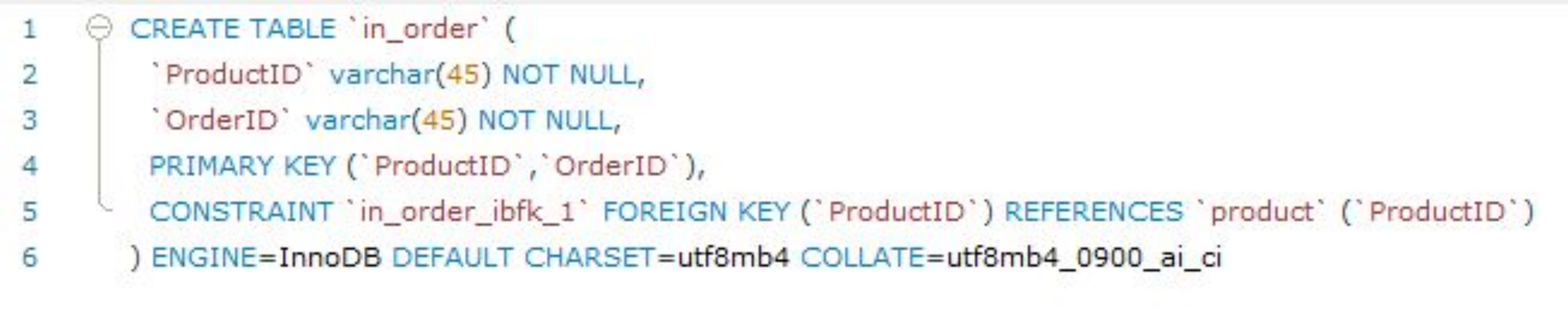
```
CREATE TABLE `customer` (
  `CustomerID` varchar(45) NOT NULL,
  `Name` varchar(45) DEFAULT NULL,
  `Gender` varchar(45) DEFAULT NULL,
  `Phone` varchar(45) DEFAULT NULL,
  `Email` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`CustomerID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Orders



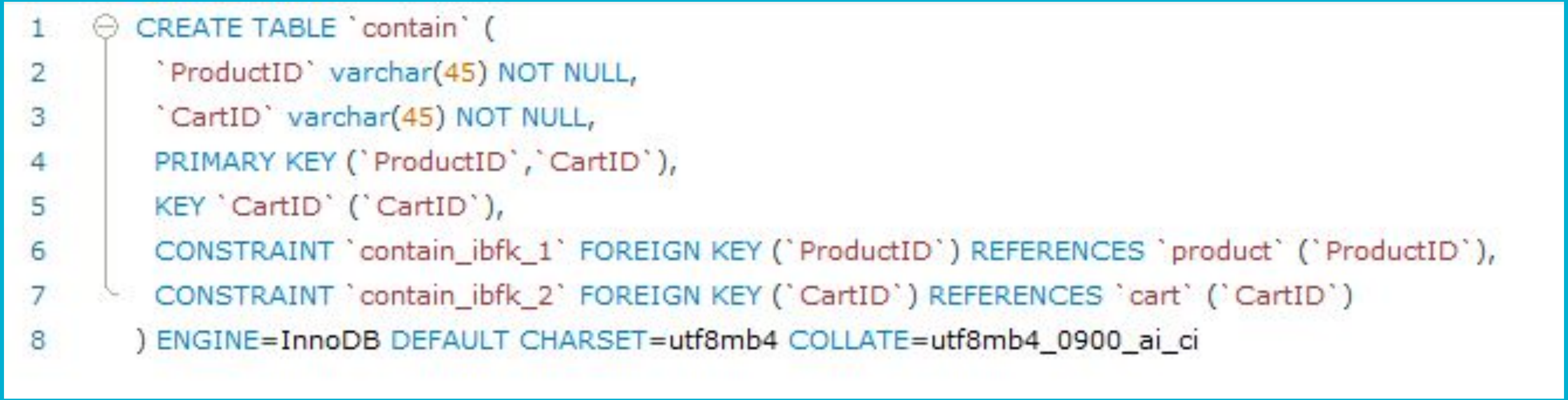
```sql
CREATE TABLE `orders` (
  `CustomerID` varchar(45) NOT NULL,
  `OrderDate` date DEFAULT NULL,
  `OrderID` varchar(45) NOT NULL,
  `Status` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`OrderID`,`CustomerID`),
  KEY `CustomerID` (`CustomerID`),
  CONSTRAINT `orders_ibfk_1` FOREIGN KEY (`CustomerID`) REFERENCES `customer` (`CustomerID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

In_Order



```
CREATE TABLE `in_order` (
  `ProductID` varchar(45) NOT NULL,
  `OrderID` varchar(45) NOT NULL,
  PRIMARY KEY (`ProductID`,`OrderID`),
  CONSTRAINT `in_order_ibfk_1` FOREIGN KEY (`ProductID`) REFERENCES `product` (`ProductID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Contain

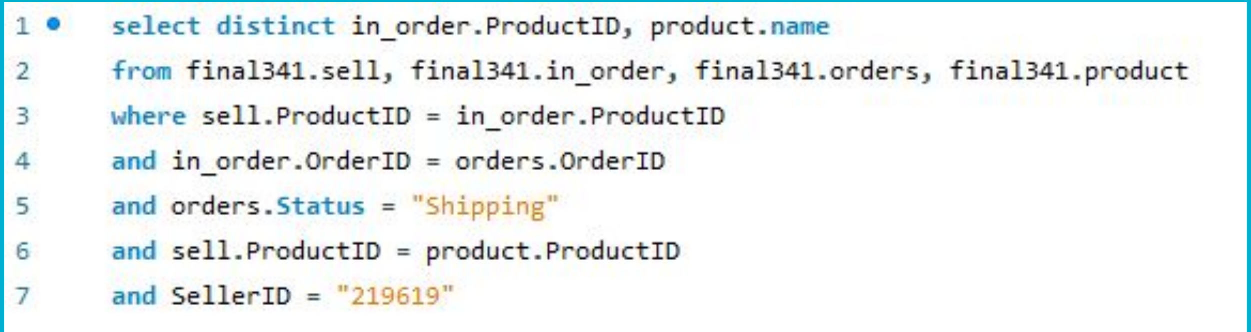


```
CREATE TABLE `contain` (
  `ProductID` varchar(45) NOT NULL,
  `CartID` varchar(45) NOT NULL,
  PRIMARY KEY (`ProductID`,`CartID`),
  KEY `CartID` (`CartID`),
  CONSTRAINT `contain_ibfk_1` FOREIGN KEY (`ProductID`) REFERENCES `product` (`ProductID`),
  CONSTRAINT `contain_ibfk_2` FOREIGN KEY (`CartID`) REFERENCES `cart` (`CartID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Sample query:

- Find all the productId and Name of the products sold by sellerId = "219619" with Status "Shipping"
- Get list of products

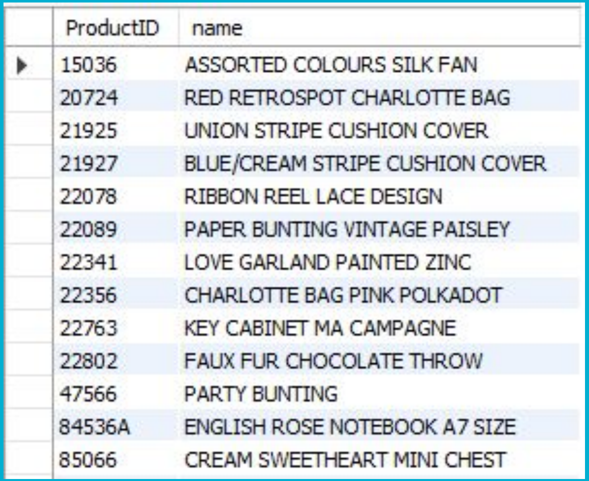
Sample query: *Find all the productId and Name of the products sold by sellerId = "219619" with Status "Shipping"*



```sql
select distinct in_order.ProductID, product.name
from final341.sell, final341.in_order, final341.orders, final341.product
where sell.ProductID = in_order.ProductID
and in_order.OrderID = orders.OrderID
and orders.Status = "Shipping"
and sell.ProductID = product.ProductID
and SellerID = "219619"
```

SQL query

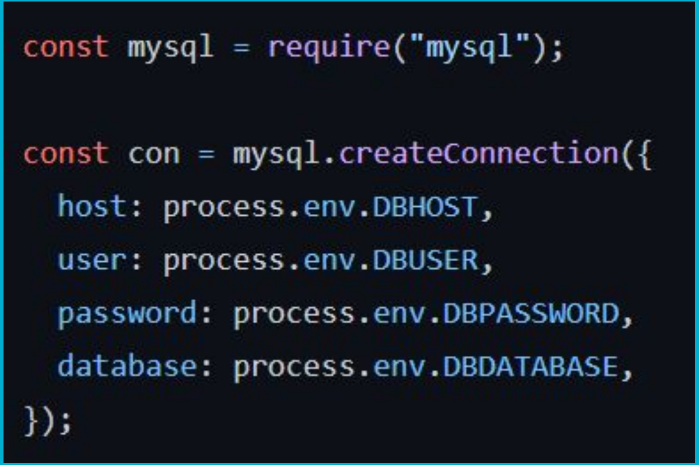
Result



ProductID	name
15036	ASSORTED COLOURS SILK FAN
20724	RED RETROSPOT CHARLOTTE BAG
21925	UNION STRIPE CUSHION COVER
21927	BLUE/CREAM STRIPE CUSHION COVER
22078	RIBBON REEL LACE DESIGN
22089	PAPER BUNTING VINTAGE PAISLEY
22341	LOVE GARLAND PAINTED ZINC
22356	CHARLOTTE BAG PINK POLKADOT
22763	KEY CABINET MA CAMPAGNE
22802	FAUX FUR CHOCOLATE THROW
47566	PARTY BUNTING
84536A	ENGLISH ROSE NOTEBOOK A7 SIZE
85066	CREAM SWEETHEART MINI CHEST

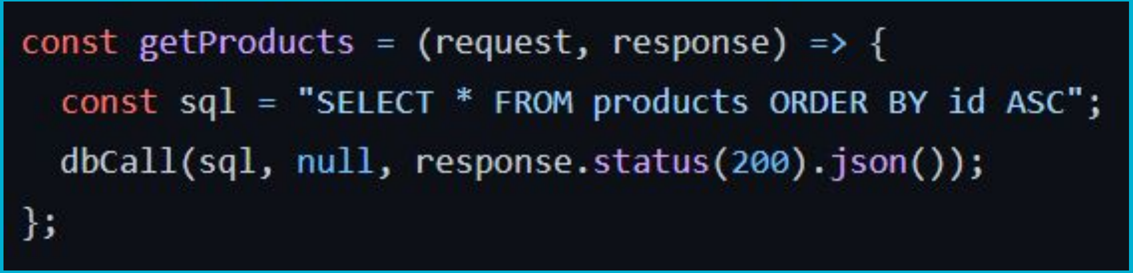
Sample query: *Get list of products*

Demo Backend's code:
con: connect to MySQL server with mysql lib
dbCall(): parse string to SQL query
getProducts(): get list of products with given sql query

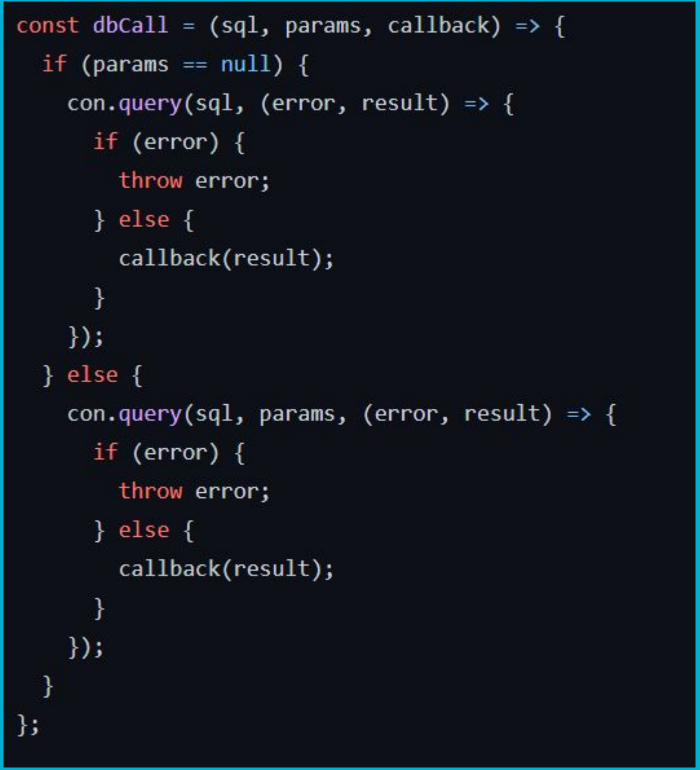


```
const mysql = require("mysql");

const con = mysql.createConnection({
  host: process.env.DBHOST,
  user: process.env.DBUSER,
  password: process.env.DBPASSWORD,
  database: process.env.DBDATABASE,
});
```



```
const getProducts = (request, response) => {
  const sql = "SELECT * FROM products ORDER BY id ASC";
  dbCall(sql, null, response.status(200).json());
};
```

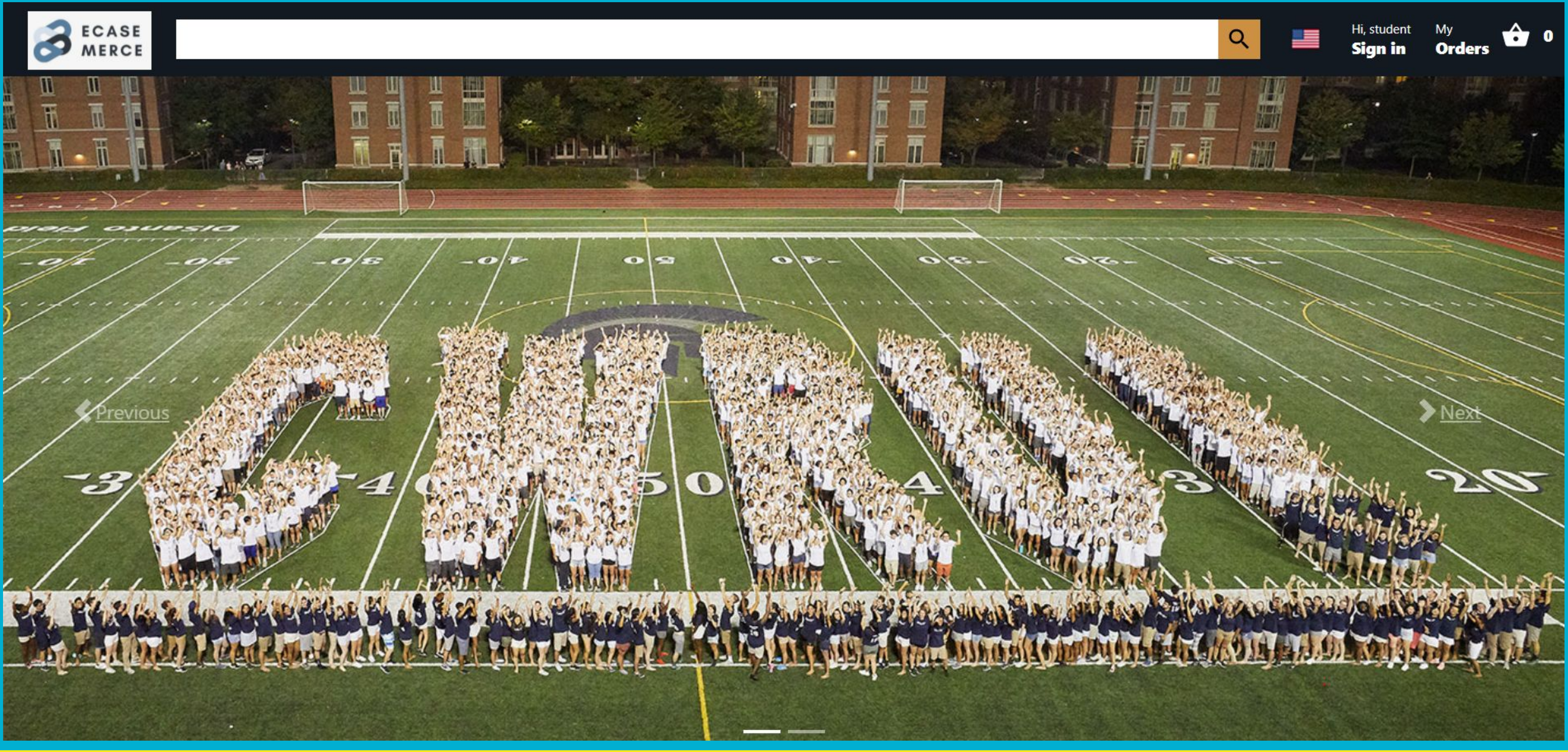


```
const dbCall = (sql, params, callback) => {
  if (params == null) {
    con.query(sql, (error, result) => {
      if (error) {
        throw error;
      } else {
        callback(result);
      }
    });
  } else {
    con.query(sql, params, (error, result) => {
      if (error) {
        throw error;
      } else {
        callback(result);
      }
    });
  }
};
```

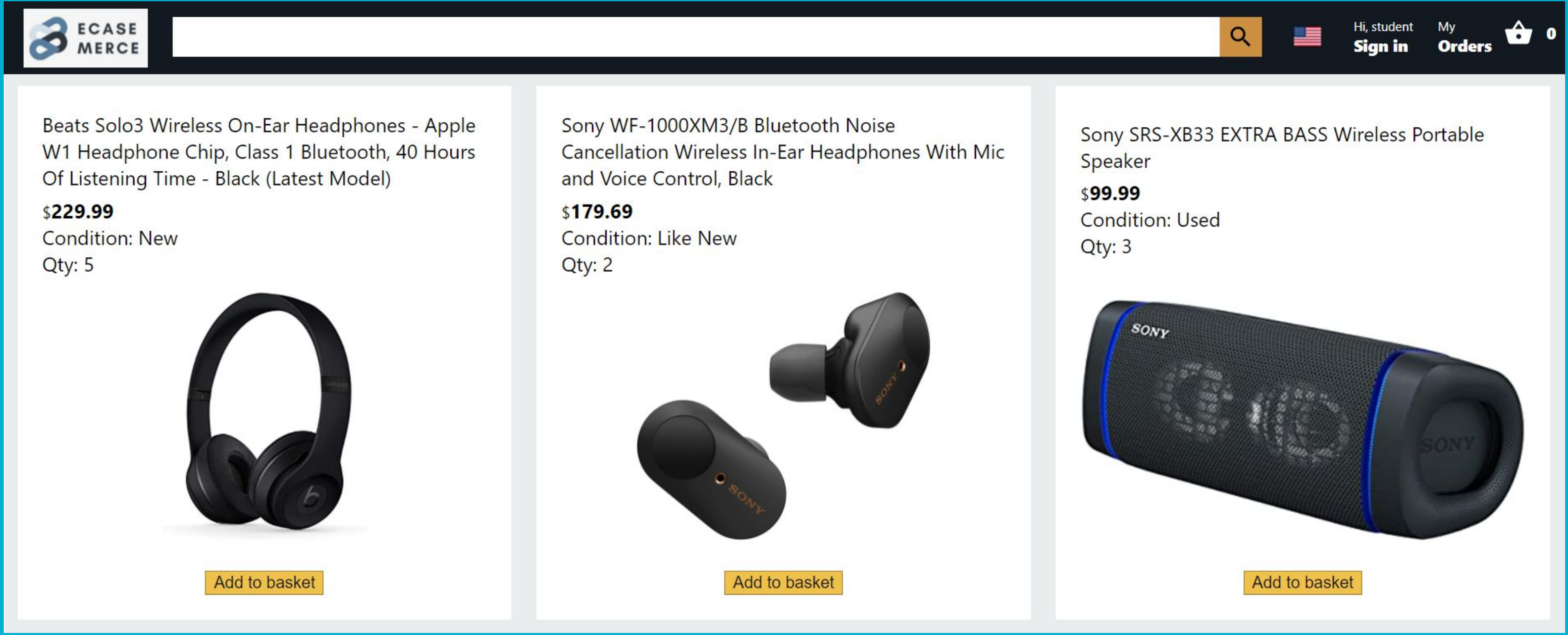
App Screenshots

- Landing Page
- Product List
- Shopping Cart

Landing Page



Product List



Beats Solo3 Wireless On-Ear Headphones - Apple W1 Headphone Chip, Class 1 Bluetooth, 40 Hours Of Listening Time - Black (Latest Model)

$**229.99**

Condition: New

Qty: 5

Sony WF-1000XM3/B Bluetooth Noise Cancellation Wireless In-Ear Headphones With Mic and Voice Control, Black

$**179.69**

Condition: Like New

Qty: 2

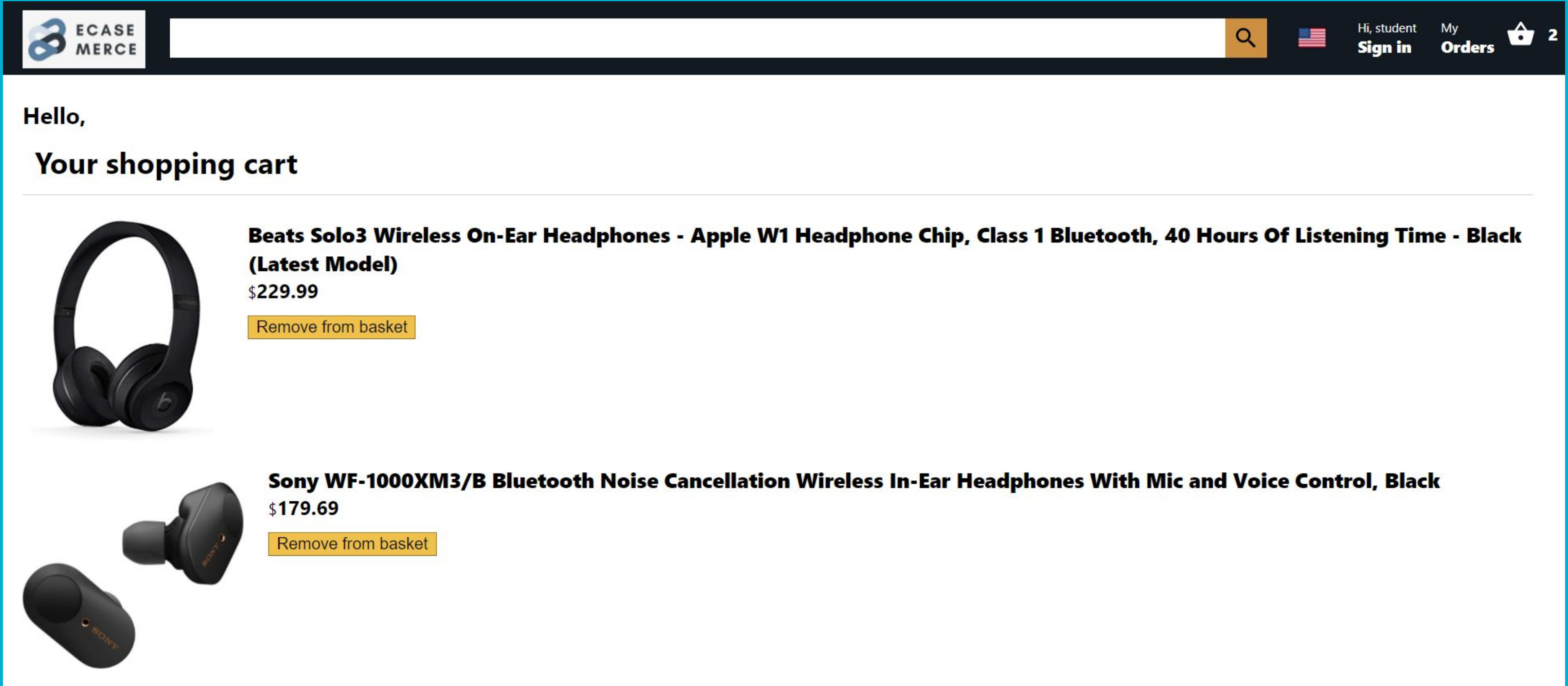
Sony SRS-XB33 EXTRA BASS Wireless Portable Speaker

$**99.99**

Condition: Used

Qty: 3

Shopping Cart



Hello,

Your shopping cart

Beats Solo3 Wireless On-Ear Headphones - Apple W1 Headphone Chip, Class 1 Bluetooth, 40 Hours Of Listening Time - Black (Latest Model)

$229.99

Sony WF-1000XM3/B Bluetooth Noise Cancellation Wireless In-Ear Headphones With Mic and Voice Control, Black

$179.69

Thank you for listening!