

# Database Course Design Report

April 2023

<b>1 INTRODUCTION .....</b>	<b>3</b>
1.1 OVERVIEW .....	3
1.2 REFERENCES .....	3
<b>2 REQUIREMENTS ANALYSIS .....</b>	<b>4</b>
2.1 DATA ANALYSIS .....	4
2.1.1 <i>Data Flow Diagram (Optional)</i> .....	
2.1.2 <i>Data dictionary</i> .....	4
2.2 FUNCTIONAL ANALYSIS .....	8
2.3 SAFETY AND INTEGRITY REQUIREMENTS .....	9
2.3.1 <i>Security Requirements</i> .....	9
2.3.2 <i>Integrity Requirement</i> .....	9
<b>3 CONCEPTUAL STRUCTURAL DESIGN .....</b>	<b>11</b>
<b>4 LOGICAL STRUCTURE DESIGN .....</b>	<b>11</b>
<b>5 PHYSICAL STRUCTURE DESIGN .....</b>	<b>15</b>
<b>6 DATABASE IMPLEMENTATION .....</b>	<b>15</b>
6.1 CREATING A DATABASE SCHEMA .....	15
6.2 DATA LOADING .....	19
6.3 APPLICATION CODING AND DEBUGGING .....	20
<b>7 DISCUSSIONS .....</b>	<b>20</b>

# Database Course Design Report

## 1 Introduction

### 1.1 Overview

**Database name:** Shopping database

**Name of database software system:** e-commerce shopping platform

**The design goal of the database is** to provide a shopping platform for consumers who have online shopping needs. Customers can browse and purchase items posted by merchants through computers or other terminals, and then form orders and shopping carts to meet their needs; administrators can manage product information, modify it, manage users, and process and view their orders. Online shopping has now become a part of daily life. People are enjoying the convenience brought by online shopping, which is incomparable to ordinary shopping. Therefore, the development of a shopping database is born to adapt to the trend of the times and meet people's needs.

**The target audience of database software system development:** consumers with online shopping needs and administrators of online shopping platforms.

**Database language:** SQL, JavaScript

### 1.2 References

【1】 Li Yajun. Design and implementation of B2C e-commerce system based on SSM framework[C]. Hefei University of Technology, 2022.

【2】 Yang Sheng, Luo Qi. Design of online shopping mall system based on Spring Boot[J]. Science and Technology Innovation and Application, 2022, 12(19): 58-61.

【3】 Chu Shulai, Zhang Pengwei. Exploration of data integrity in SQL Server[J]. Computer CD-ROM Software and Applications, 2013, 16(05): 185+187.

## 2. Demand Analysis

### 2.1 Data analysis

#### 2.1.2 Data Dictionary

##### a . Document List

Document number	file name	Xiaomi Computer
1	Processing and Archiving Requirements	Crawling <a href="https://www.mi.com/">https://www.mi.com/</a> through python crawler statements Import computer-related data from the web page into Excel tables by category
	Data item list	product_id , commodity_id, goods_id, name, price, market_price, reduce, buy_limit, image_url
Document number	file name	Xiaomi mobile phone
2	Processing and Archiving Requirements	Crawling <a href="https://www.mi.com/">https://www.mi.com/</a> through python crawler statements Import mobile phone-related data from the web page into Excel tables by category
	Data item list	product_id , commodity_id, goods_id, name, price, market_price, reduce, buy_limit, image_url
Document number	file name	flat
3	Processing and Archiving Requirements	Crawling <a href="https://www.mi.com/">https://www.mi.com/</a> through python crawler statements Import the tablet-related data from the web page into the Excel table by category
	Data item list	product_id , commodity_id, goods_id, name, price, market_price, reduce, buy_limit, image_url

Document number	file name	List of orders
4	Processing and Archiving Requirements	Collect relevant order data and filter and sort it
	Data item list	Order ID , Order Date, Customer Name, State, City

## **b . List of local data items**

Local serial number	Data item name		Global sequence number	Data item name		Corresponding document number
	English	Chinese		English	Chinese	
	product_id	Product Number				1
	name	product name				1
	price	price				2
	image_url	image link				3
	Order ID	Order Number				4
	CustomerName	username				4

## **c . List of data items**

Serial number	Data item name	meaning	type	length	Ranges	other
1	user_id	user ID	Int		Not null	
2	Username	user name	Verc	45		

			har			
3	Nickname	User's Nickname	Verc har	45		
4	Password	User login password	Verc har	255		
5	Sex	User Gender	Tinyint	1		
6	Age	User age	Int			
7	Address	User Address	Verc har	45		
8	User_pic	profile picture	Medium blob			
9	Status	user status	Tinyint			
10	Email	User mailbox	Verc har	45		
11	Manager_id	Administrator Number	Int		Not null	
12	Manager_name	Administrator Name	Verc har	45		
13	Manager_password	Administrator Pass	Verc har	255		
14	Products_id	Product Number	Int		Not null	
15	Imgs	product picture	Text			
16	Name	Product Name	Text			
17	Price	Product Prices	Int			
18	Category_id	Classification Number	Int		Not null	
19	Inventory	stock	Int			

20	Category_name	Category Name	Varchar	45		
twenty one	Order_num	The quantity of a product in an order	Int			
twenty two	Order_id	Order Number	Int		Not null	
twenty three	Order_status	Order existence status	Tinyint			
twenty four	Payment	Order payment status	Varchar	45		
25	Cart_id	Shopping cart number	Int		Not null	

## d . Data Structure List

Serial number	Data structure name	meaning	composition
1	User	User Info	User ID, user name, user address, user gender, user nickname, password, age, email address, whether the account has been cancelled
2	Manager	Administrator Information	Administrator ID, Administrator Name
3	Products	Product Information	Product number, product name, price, picture, whether it has been removed from the shelves
4	Categories	Product classification information	Category number, whether deleted, category name
5	Orders	order information	Order number, product quantity, user number, product number, whether paid
6	Carts	Shopping cart information	User ID, Product Quantity, Product ID

## 2.2 Functional analysis

### a. Functional division

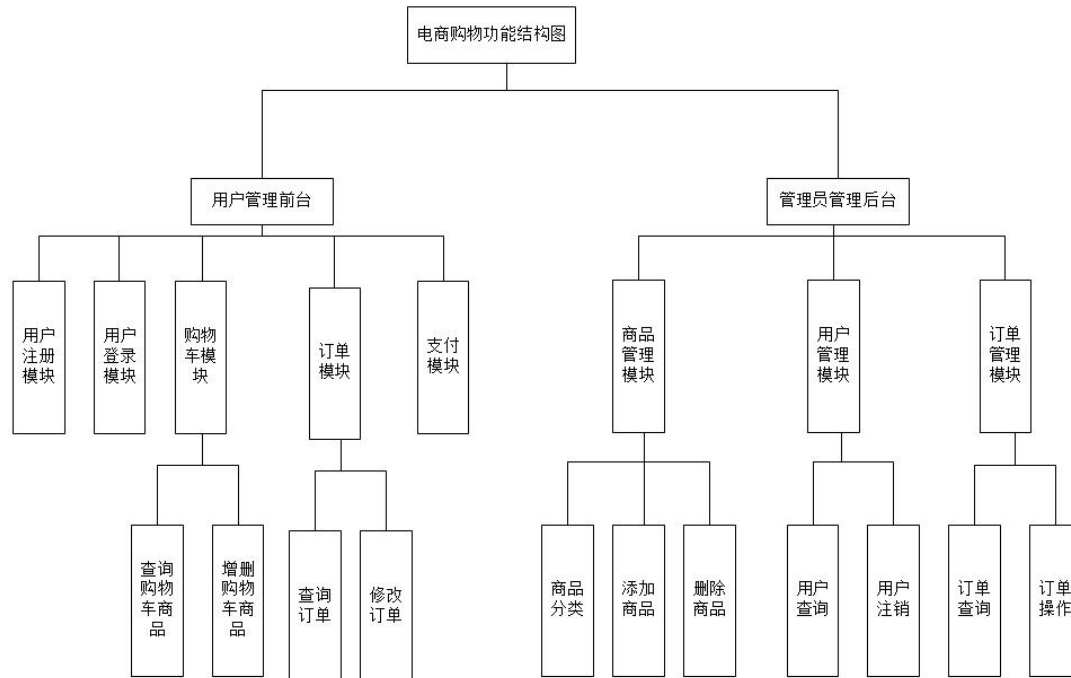


Figure 1 System structure and function diagram

**The function of e-commerce shopping database is** to store table data such as users, products, and orders.

**The function of the application system:** to operate on data, such as querying shopping cart items, modifying orders, etc.; to request data from the database; to respond to data operations and display the results of data operations, such as displaying shopping cart items, etc.

#### **User management front desk:**

User management: User management includes user registration module, user login module, shopping cart module, product classification module, order module, and payment module functions. User management is mainly for user services.

User registration module: Users fill in relevant information on the registration page. For example, a user can only register one account, and one account corresponds to one user; unregistered users are visitors and can only browse product pages but cannot perform other operations.

User login module: Users can log in to the system by entering their registered ID and the corresponding password. After logging in, they can browse the product page, purchase products, pay orders, etc.

Shopping cart module: The shopping cart page can display the quantity added by the user. The user can view the items in his or her shopping cart and add or delete items from the shopping cart.

Order module: The order page can display all added product information, product unit price and payment status. Users can query the product information of the order or delete the order;

Payment module: Payment means that users can settle orders. After the payment is settled, the order status changes from pending payment to paid status.



**Administrator management backend:**

Administrator management: Administrator management includes product management module, user management module, and order management module functions.

Administrator management is mainly for administrators.

Product management module: It has the function of classifying products into different categories; it can add a new product to the database ; and it can delete products from the product table;

User management module: User query function, which can view the information of registered users and modify it; user logout function, which can log out the user's account and delete all the user's information from the user table;

Order management module: Administrators can query the orders that users have paid for, and can view the user's order information, order payment status, and order amount.

## 2.3 Safety and integrity requirements

### 2.3.1 Security Requirements

**User identity authentication** : To ensure system security, certain operations must be performed after logging in. Customers and administrators can only browse product information before logging in, and cannot perform other operations. After user registration, an account ID and password will be created, through which the user can log in to purchase products and other operations. Administrators will also have different permissions and functions after logging in.

**Autonomous access control method** : Users and administrators have different permissions. Administrators are authorized to delete products in the product table, but users cannot. In order to ensure the security of system operation and prevent unauthorized access and malicious operations, it is necessary to establish a role-based access control (RBAC) strategy. According to the functional requirements shown in the functional analysis section, different permissions are assigned to the two different roles of administrator and customer. When a user logs in to the system, for each operation initiated by the user, the system will first determine whether the role corresponding to the current user has the specified operation permission. If the user has the corresponding operation permission, the current operation is allowed to be executed; otherwise, the current user is prohibited from executing the operation and is forced to exit the system. [2]

### 2.3.2 Integrity requirements

#### 1. Entity integrity constraints

Entity integrity requires that each row of data in the table reflects a different entity, and duplicate data rows cannot exist. Entity integrity in the table can be achieved through indexes, unique

constraints, primary key constraints, or marker column attributes.

## 2. Domain integrity constraints

Domain integrity refers to the validity of the input of a given column. Domain integrity of a table can be achieved through various methods such as limiting data types, check constraints, input formats, foreign key constraints, default values, non-null constraints, etc.

## 3. Referential integrity constraints

Referential integrity constraints maintain defined relationships between tables when rows of data are entered or deleted.

Users are prohibited from implementing the following functions: (1) Adding records to a child table when there are no related records in the primary table ; (2) Changing values in the primary table and causing orphaned records in the child table. (3) Deleting a record from the primary table, but there are still related records in the child table that match the record. Referential integrity is achieved through the reference relationship between the primary key and the foreign key.

## 4. Custom integrity constraints

User-defined integrity is used to define specific rules. If a user has requirements for a certain attribute, the rule can be implemented through statements without violating entity integrity and referential integrity. [3]

### 3 Conceptual structure design



Figure 2 Global ER diagram

### 4 Logical structure design

Users: (user\_id, username, nickname, password, sex, age, address, user\_pic, status, email)

The user's primary attribute is user\_id

Managers: (manager\_id, manager\_name, manager\_password, status)

The primary attribute of an administrator is manager\_id

Products: (product\_id, name, price, imgs, category\_id, status, inventory)

The primary attribute of a product is product\_id, and the foreign key is category\_id

Catalogs : (category\_id,category\_name,status)

The main attribute of the catalog is category\_id

Orders: (order\_id, product\_id, user\_id, order\_num, order\_status, payment)

The primary attribute of an order is order\_id, and the foreign key is product\_id, user\_id

Shopping carts: (cart\_id, user\_id, product\_id, product\_num)

The shopping cart keyword is cart\_id, and the foreign keywords are product\_id, user\_id

#### List of Tables

Serial number	Table Name	illustrate
1	users	User, store user information
2	managers	Administrator, stores administrator information
3	products	Products, store information about goods
4	Category	Classification, storing information about product classification
5	orders	Orders, which store information about items purchased by users
6	carts	Shopping cart, which stores information about products that users intend to purchase

Table 1 : users

Column Name	Data Type (Length)	PK / FK	meaning	Value range and constraints
user_id	int	P K	User ID	not null
username	varchar( 45 )		username	
nickname	varchar( 45 )		User's Nickname	
Password	varchar( 255 )		User Pass	
sex	tinyint(1)		User Gender	
age	int		User age	
address	varchar( 45 )		User Address	
status	tinyint		User account status	
email	varchar( 45 )		User mailbox	
user_pic	mediablob		user password	

Table 2 : managers

Column Name	Data Type (Length)	PK / FK	meaning	Value range and constraints
manager_id	int	PK	Administrator ID	not null

manager_name	varchar(45)		Administrator Name	
manager_password	varchar(255)		Administrator Account	
status	tinyint		Administrator status	

Table 3 : products

Column Name	Data Type (Length)	PK / FK	meaning	Value range and constraints
product_id	int	PK	Product ID	not null
name	text		product name	
price	int		Product Prices	
imgs	text		Product page	
category_id	int	P K	Product category id	not null
inventory	int			
status	tinyint		Product existence status	

Table 4 : category

Column Name	Data Type (Length)	PK / FK	meaning	Value range and constraints
category_id	int	PK	Product category id	not null
category_name	varchar( 45 )		Product Category Name	
status	tinyint		Product classification status	

Table 5 : orders

Column Name	Data Type (Length)	PK / FK	meaning	Value range and constraints
order_id	int	PK	Order ID	not null
product_id	int		Product ID	
user_id	int		User ID	
order_num	int		Number of	

			Products	
order_status	tinyint		Order existence status	
Payment	varchar(45)		Order payment status	

Table 6 : carts

Column Name	Data Type (Length)	PK / FK	meaning	Value range and constraints
car_id	int	PK	Shopping cart number	not null
user_id	int		User ID	
product_id	int		Product ID	
product_num	int		Number of Products	

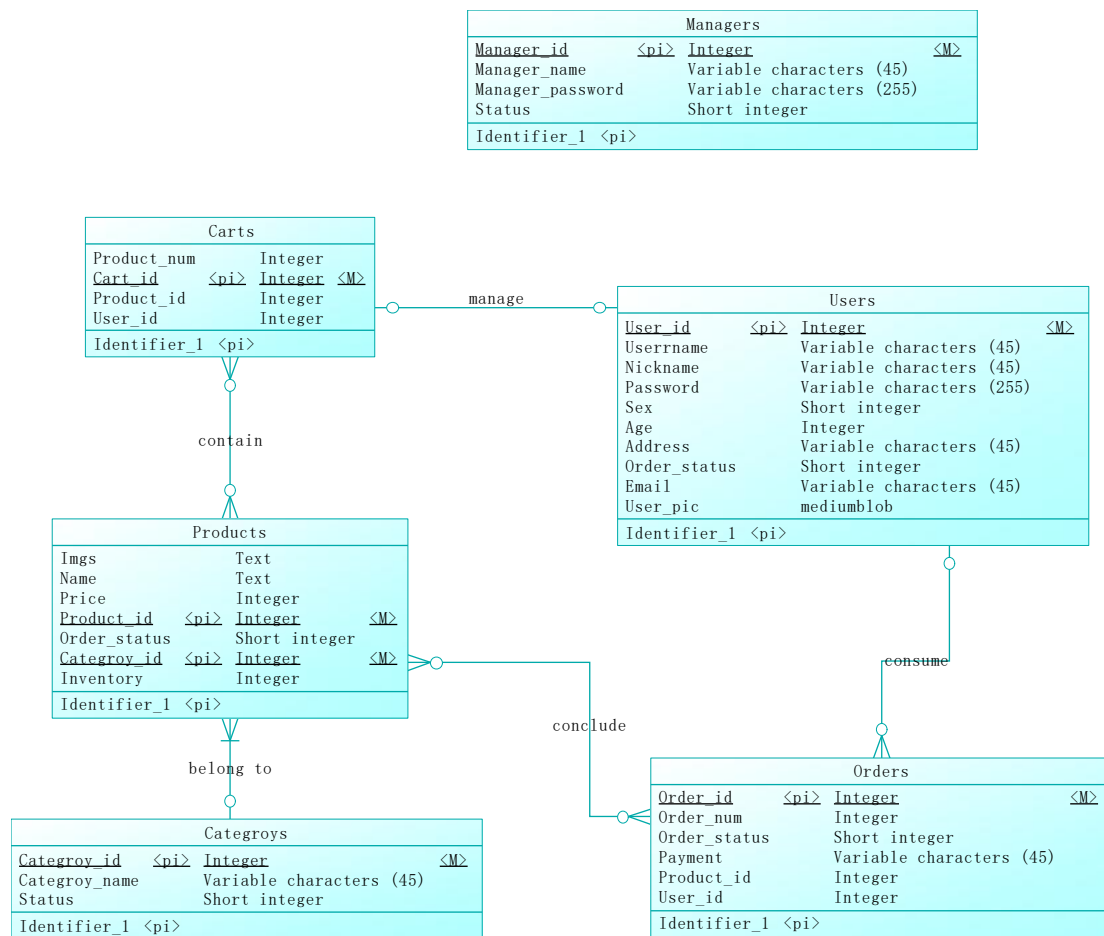


Figure 3 Logical structure design diagram

## 5 Physical structure design

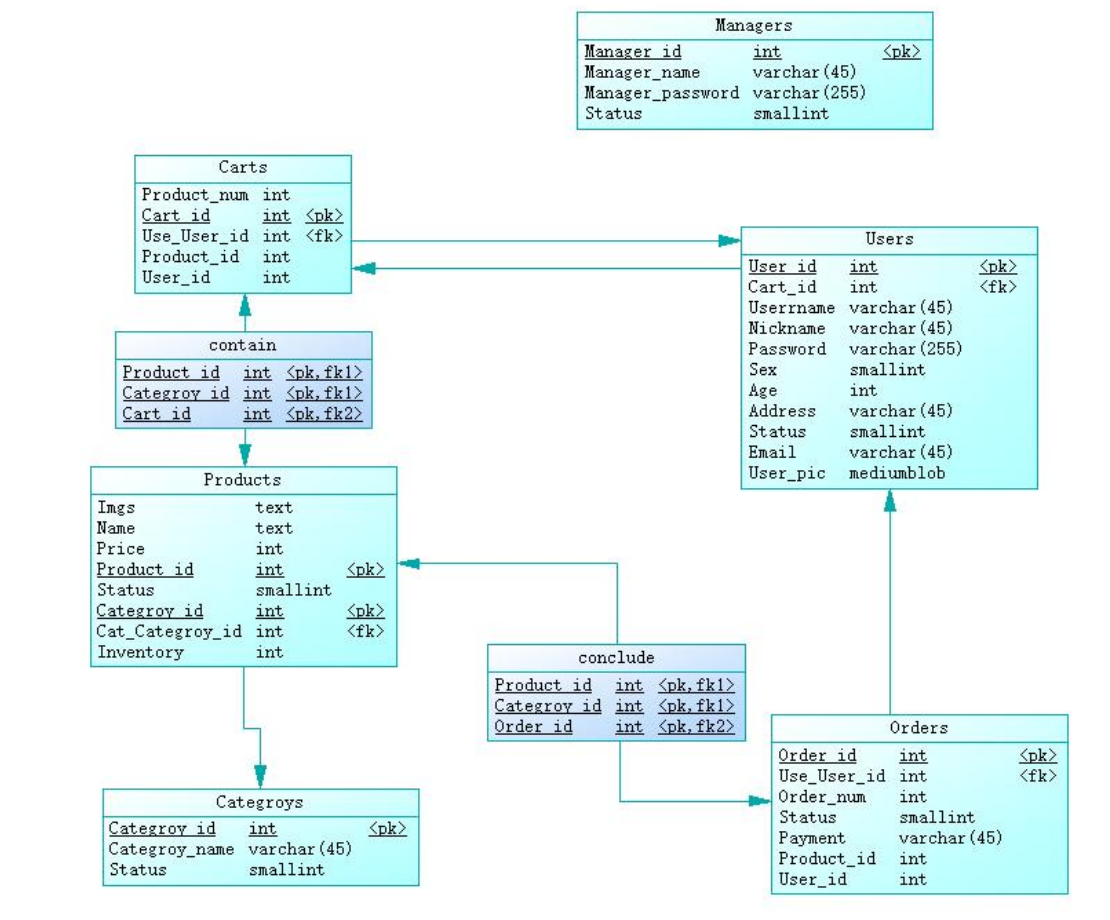


Figure 4 Physical structure design diagram

## 6 Database Implementation

Use the database language and high-level language provided by the database management system to establish a database based on the results of logical structure design and physical structure design, write and debug application programs, organize data into the database, and conduct trial runs.

### 6.1 Create Database Schema

Database scripts

drop table if exists CARTS;

drop table if exists CATEGROYS;

drop table if exists MANAGERS;

drop table if exists ORDERS;

drop table if exists PRODUCTS;

drop index charge\_FK on USERS;

drop table if exists USERS;

drop table if exists conclude;

drop table if exists contain;

```
/*=====
/* Table: CARTS */
/*=====
create table CARTS
(
  PNUM          int,
  CID2          int not null,
  USE_UID       int,
  PID          int,
  UID          int,
  primary key (CID2)
);
```

```
/*=====
/* Table: CATEGROYS */
/*=====
create table CATEGROYS
(
  CID          int not null,
  CNAME        varchar(45),
  STA          smallint,
  primary key (CID)
);
```

```
/*=====
/* Table: MANAGERS */
/*=====
create table MANAGERS
(
  MID          int not null,
  MNAME        varchar(45),
  MPAW        varchar(255),
  STA          smallint,
  primary key (MID)
);
```



```

/*=====
/* Table: MANAGERS                                     */
/*=====
create table MANAGERS
(
  MID          int not null,
  MNAME        varchar(45),
  MPAW         varchar(255),
  STA          smallint,
  primary key (MID)
);

/*=====
/* Table: ORDERS                                       */
/*=====
create table ORDERS
(
  OID          int not null,
  USE_UID      int,
  ONUM         int,
  STATUS       smallint,
  PAY          varchar(45),
  PID          int,
  UID          int,
  primary key (OID)
);

/*=====
/* Table: PRODUCTS                                     */
/*=====
create table PRODUCTS
(
  PIMGS        text,
  PNAME        text,
  PPRICE       int,
  PID          int not null,
  STATUS       smallint,
  CID          int not null,
  CAT_CID      int,
  INV          int,
  primary key (PID, CID)
);

/*=====
/* Table: USERS                                       */
/*=====
create table USERS
(
  UID          int not null,
  CID2         int,
  UNAME        varchar(45),
  NNAME        varchar(45),
  PNO          varchar(255),
  USEX         smallint,
  UAGE         int,
  UADDRESS     varchar(45),
  STATUS       smallint,
  UEMAIL       varchar(45),
  UPIC         mediumblob,
  primary key (UID)
);

```

```

/*=====
/* Index: charge_FK                                */
/*=====
create index charge_FK on USERS
(
);

/*=====
/* Table: conclude                                */
/*=====
create table conclude
(
    PID          int not null,
    CID          int not null,
    OID          int not null,
    primary key (PID, CID, OID)
);

/*=====
/* Table: contain                                */
/*=====
create table contain
(
    PID          int not null,
    CID          int not null,
    CID2         int not null,
    primary key (PID, CID, CID2)
);

alter table CARTS add constraint FK_manage foreign key (USE_UID)
references USERS (UID) on delete restrict on update restrict;

alter table ORDERS add constraint FK_consume foreign key (USE_UID)
references USERS (UID) on delete restrict on update restrict;

alter table PRODUCTS add constraint "FK_belong to" foreign key (CAT_CID)
references CATEGROYS (CID) on delete restrict on update restrict;

alter table USERS add constraint FK_manage2 foreign key (CID2)
references CARTS (CID2) on delete restrict on update restrict;

alter table conclude add constraint FK_conclude foreign key (PID, CID)
references PRODUCTS (PID, CID) on delete restrict on update restrict;

alter table conclude add constraint FK_conclude2 foreign key (OID)
references ORDERS (OID) on delete restrict on update restrict;

alter table contain add constraint FK_contain foreign key (PID, CID)
references PRODUCTS (PID, CID) on delete restrict on update restrict;

alter table contain add constraint FK_contain2 foreign key (CID2)
references CARTS (CID2) on delete restrict on update restrict;

```

## 6.2 Data loading

Commodities are imported using a .csv table, and user, category, order, order and other related information are added using related interfaces to input data, such as:

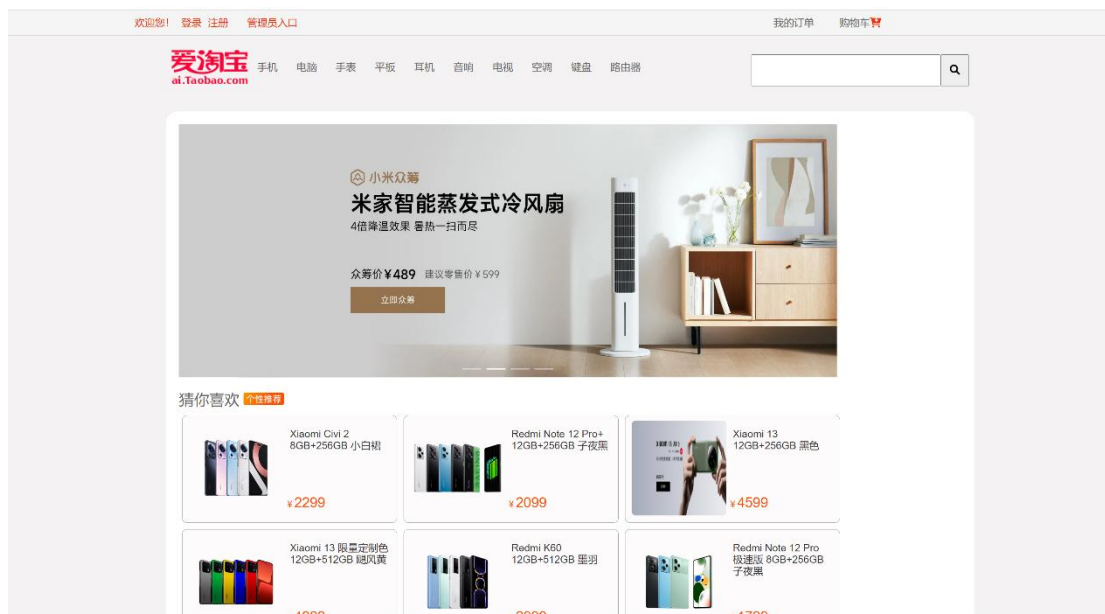
product_id	iname	price	imgs
17971	Xiaomi 13	4599	<a href="https://cdn.cnbj0.fds.api.mi-img.com/b2c-shopapi-pms/pms_1670745532.86084186.png">https://cdn.cnbj0.fds.api.mi-img.com/b2c-shopapi-pms/pms_1670745532.86084186.png</a>
16536	Xiaomi Civ	2299	<a href="https://cdn.cnbj0.fds.api.mi-img.com/b2c-shopapi-pms/pms_1664192126.84785883.png">https://cdn.cnbj0.fds.api.mi-img.com/b2c-shopapi-pms/pms_1664192126.84785883.png</a>
18075	Redmi K60	2999	<a href="https://cdn.cnbj0.fds.api.mi-img.com/b2c-shopapi-pms/pms_1672220460.25837153.png">https://cdn.cnbj0.fds.api.mi-img.com/b2c-shopapi-pms/pms_1672220460.25837153.png</a>
17972	Xiaomi 13	4999	<a href="https://cdn.cnbj0.fds.api.mi-img.com/b2c-shopapi-pms/pms_1670746082.9754133.png">https://cdn.cnbj0.fds.api.mi-img.com/b2c-shopapi-pms/pms_1670746082.9754133.png</a>
18268	Redmi Note	2099	<a href="https://cdn.cnbj0.fds.api.mi-img.com/b2c-shopapi-pms/pms_1679982568.10982510.png">https://cdn.cnbj0.fds.api.mi-img.com/b2c-shopapi-pms/pms_1679982568.10982510.png</a>
18092	Redmi Note	1799	<a href="https://cdn.cnbj0.fds.api.mi-img.com/b2c-shopapi-pms/pms_1672039203.85771220.png">https://cdn.cnbj0.fds.api.mi-img.com/b2c-shopapi-pms/pms_1672039203.85771220.png</a>
18119	Redmi 12C	669	<a href="https://cdn.cnbj0.fds.api.mi-img.com/b2c-shopapi-pms/pms_1672368924.1898024.png">https://cdn.cnbj0.fds.api.mi-img.com/b2c-shopapi-pms/pms_1672368924.1898024.png</a>
16536	Xiaomi Civ	2299	<a href="https://cdn.cnbj0.fds.api.mi-img.com/b2c-shopapi-pms/pms_1664192126.84785883.png">https://cdn.cnbj0.fds.api.mi-img.com/b2c-shopapi-pms/pms_1664192126.84785883.png</a>
18268	Redmi Note	2099	<a href="https://cdn.cnbj0.fds.api.mi-img.com/b2c-shopapi-pms/pms_1679982568.10982510.png">https://cdn.cnbj0.fds.api.mi-img.com/b2c-shopapi-pms/pms_1679982568.10982510.png</a>
16646	Redmi Note	2099	<a href="https://cdn.cnbj0.fds.api.mi-img.com/b2c-shopapi-pms/pms_1666797744.54085396.png">https://cdn.cnbj0.fds.api.mi-img.com/b2c-shopapi-pms/pms_1666797744.54085396.png</a>

**POST** 管理员注册 > **POST** 增加分类名称 (1) **POST** 管理员增加商品

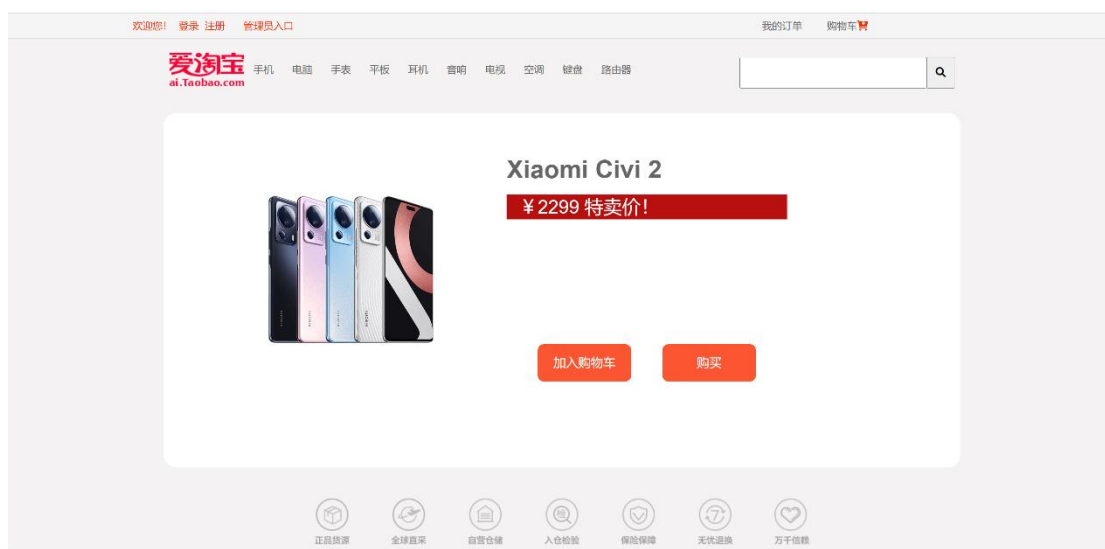
> **POST** 将商品添加到购物车 (2)

## 6.3 Application coding and debugging

Home page after opening:



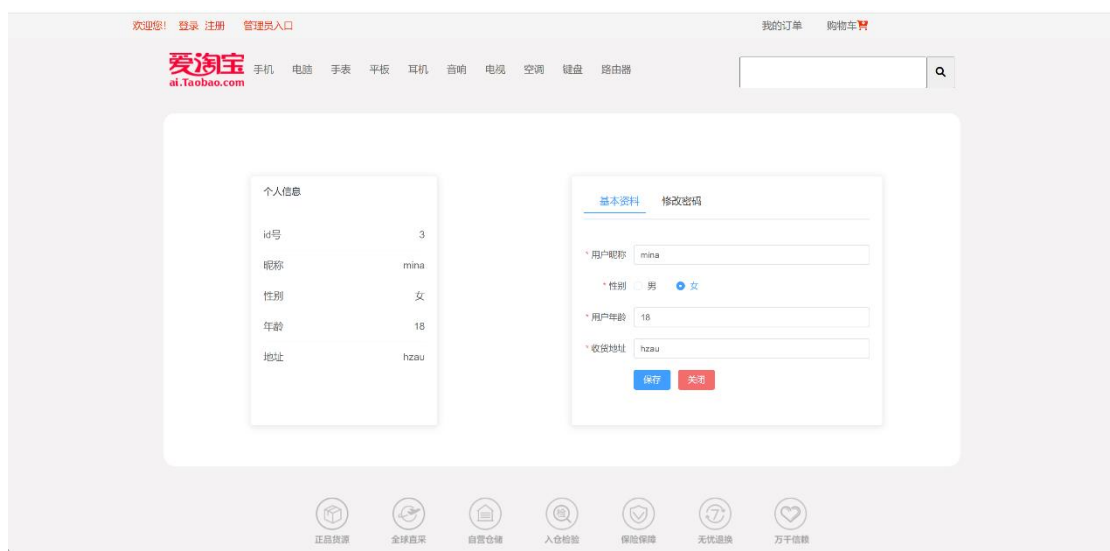
You can click on the product to view the product details:



Click Add to Cart and you will be automatically redirected to the login page. There is also a login option on the homepage.



After logging in, you will be redirected to the homepage. You can add items to the shopping cart, purchase items, view your personal center, and click "Welcome" to enter your personal center:



In the personal center, you can modify your personal information and password:

基本资料

修改密码

\* 旧密码

请输入旧密码

\* 新密码

请输入新密码

\* 确认密码

请确认新密码

保存

关闭

Click My Order to view the user's own orders:

爱淘宝

手机 电脑 手表 平板 耳机 音响 电视 空调 键盘 路由器

我的订单

购物车

所有订单

待付款

已付款

宝贝	单价	数量	交易状态
<div>2023-02-20 订单号: 7</div> <div> <div> <div>Xiaomi Civi 2</div> <div>8GB+256GB 小白樱</div> </div> </div>			
¥ 2299	2	已付款	
<div>2023-02-20 订单号: 8</div> <div> <div> <div>Redmi Note 12 Pro+</div> <div>12GB+256GB 子夜黑</div> </div> </div>			
¥ 2099	1	<div>点击付款</div> <div>待支付</div>	
<div>2023-02-20 订单号: 9</div> <div> <div> <div>Redmi Note 12 Pro 极速版</div> <div>8GB+256GB 子夜黑</div> </div> </div>			
¥ 1799	2	<div>点击付款</div> <div>待支付</div>	

All orders will appear first, including unpaid and paid orders. Click on Pay and the order will change from unpaid to paid:

2023-02-20 订单号: 8

Redmi Note 12 Pro+

12GB+256GB 子夜黑

¥ 2099

1

点击付款

待支付

After clicking



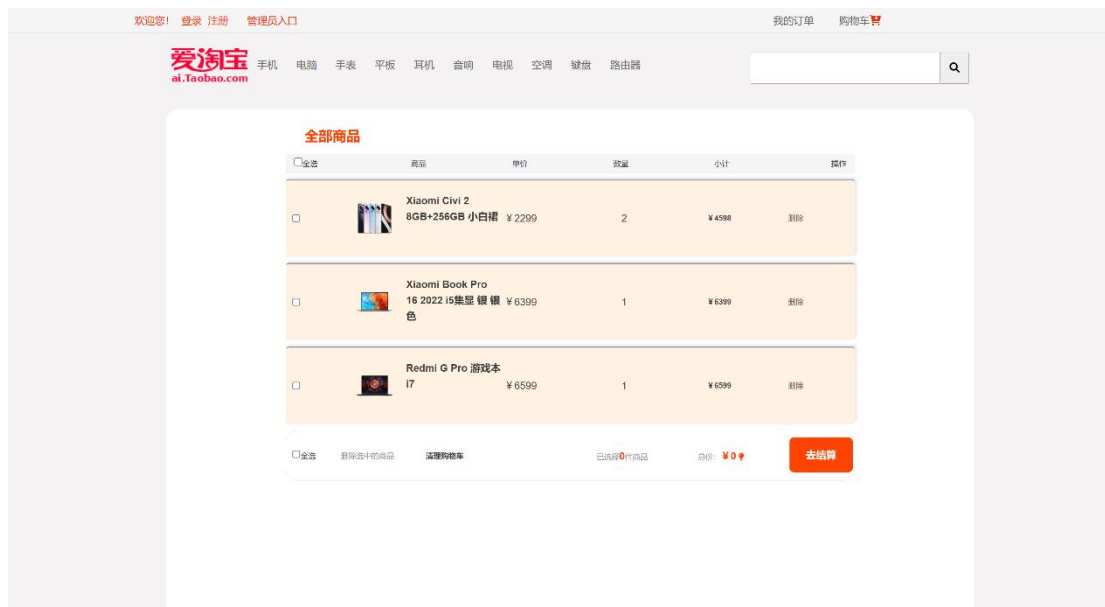
Redmi Note 12 Pro+  
12GB+256GB 子夜黑

¥2099

1

已付款

You can check the different status of your order by clicking on Paid and Unpaid above.  
shopping cart:

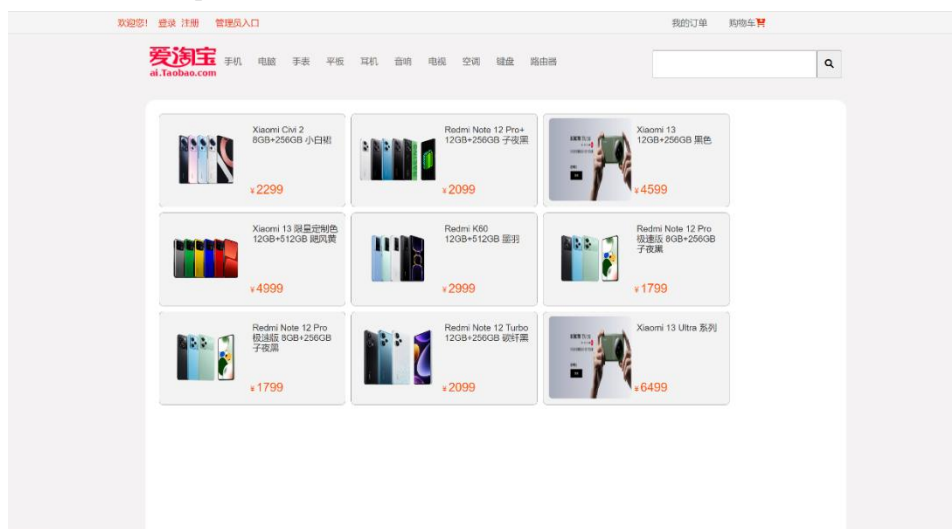


The shopping cart only has the delete function completed.

Click the logo to jump to the homepage, and click the relevant category name to view the products in that category:

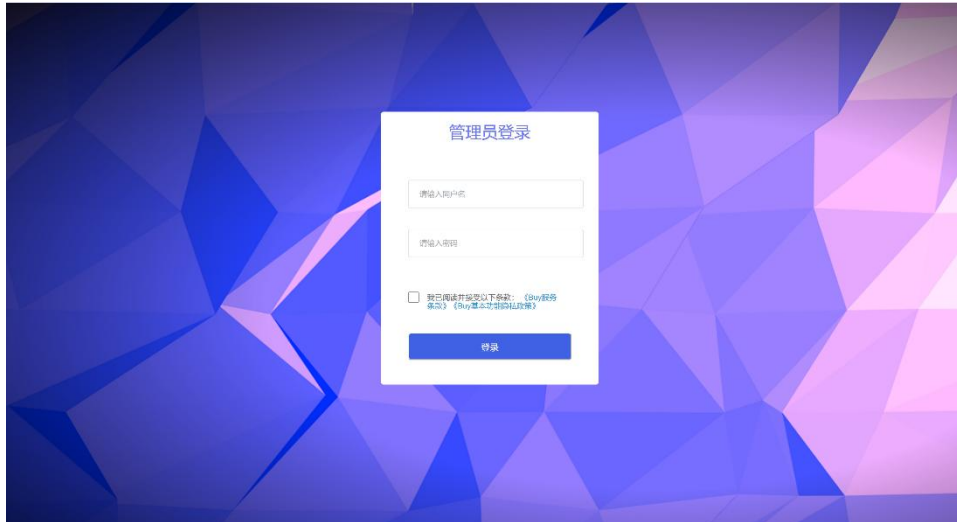


Click on mobile phone:



This page can also be used to view product details and perform operations on products.  
Click the administrator entrance to log in as an administrator:





After logging in, you can perform administrator-related operations. (No page is displayed for this part)

The following operations can be performed by administrators:

▼	manager (15)	
▼	products (4)	
	POST 管理员真正删除一个商品	•
	POST 管理员增加商品	•
	GET 管理员获取所有商品数据	•
	POST 管理员修改商品数据	•
▼	categorys (5)	
	POST 获取分类名称	•
>	POST 增加分类名称 (1)	•
	POST 删除分类	•
	POST 根据id获取分类	•
>	POST 更改分类名称 (1)	•
▼	users (2)	
	GET 管理员获取全部用户的信息	•
	POST 删除已经注销的用户	•
▼	orders (1)	
	GET 获取所有订单信息	•
▼	carts (1)	
	GET 获取所有购物车信息	•
>	POST 管理员登陆 (1)	•
	POST 管理员注册	•

## 7 Discussions

The main business processes of e-commerce involve products, users, orders, etc., and multiple tables are generated according to the processes. After discussion and communication, our group created six tables: shopping cart, user, order, product catalog, administrator, and product, and determined the relevant functions during the discussion and communication process.