**INFSCI 2710: DATABASE MANAGEMENT SYSTEM**

**PROJECT REPORT**

**DIGITAL PRODUCT SALES MANAGEMENT SYSTEM**

**University of Pittsburgh**

**By:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Student ID** | **Email Address** |
| 1 | Huanjie Dong | 4301180 | hud8@pitt.edu |
| 2 | Haohan Zhong | 4374244 | haz98@pitt.edu |
| 3 | Ying Zhang | 4379324 | yiz165@pitt.edu |

**Contents**

[1. Overview - 3 -](#_Toc26281518)

[2. Assumptions - 3 -](#_Toc26281519)

[3. ER Diagram - 3 -](#_Toc26281520)

[4. Database Design - 4 -](#_Toc26281521)

[5. Connection between Front-end and Back-end - 9 -](#_Toc26281522)

[6.  System Implementation - 10 -](#_Toc26281523)

[7. Erroneous Cases Handled - 11 -](#_Toc26281524)

[8. Limitations and Improvements - 11 -](#_Toc26281525)

# 1. Overview

Digital products sales system has a list of digital products. They are categorized by laptops, phones, cameras, headphones and TV. Customers can browse and search on the website, and buy one product each time. The application will have two users. They are explained as follows:

1) Users/Customers: People who will purchase products.

a) Home customers: Users with more attributes, such as user name, password, marriage, gender, age, income, address, category

b) Business customers: Users with less attributes, such as user name, password, income, category

All users will be able to login into the system if they have their user-id and password created. After logging in they will see a home page with product attributes and 3 new arrivals.

# 2. Assumptions

1) Users must enter all information before registering, no empty information

2) Users can only purchase one item for an order at a time

3) Salespersons are independent of each other, and each order is randomly assigned a salesperson

4) Back-end database can only be opened by management staff, modifying data in tables and product information

# 3. ER Diagram

As shown in the figure below, the primary key is a bold and underlined field in the ER diagram. The foreign keys are in the same name with connected form, except “salesperson\_name” in the “store” form and “product\_id” in the “transaction” form.

As for the relationships between forms, one special case is the customer form. In order to achieve different functions of customers like business customers and home customers, we create two different forms use different attributes, and connect them to “customer” form with the foreign key “customer\_id”.

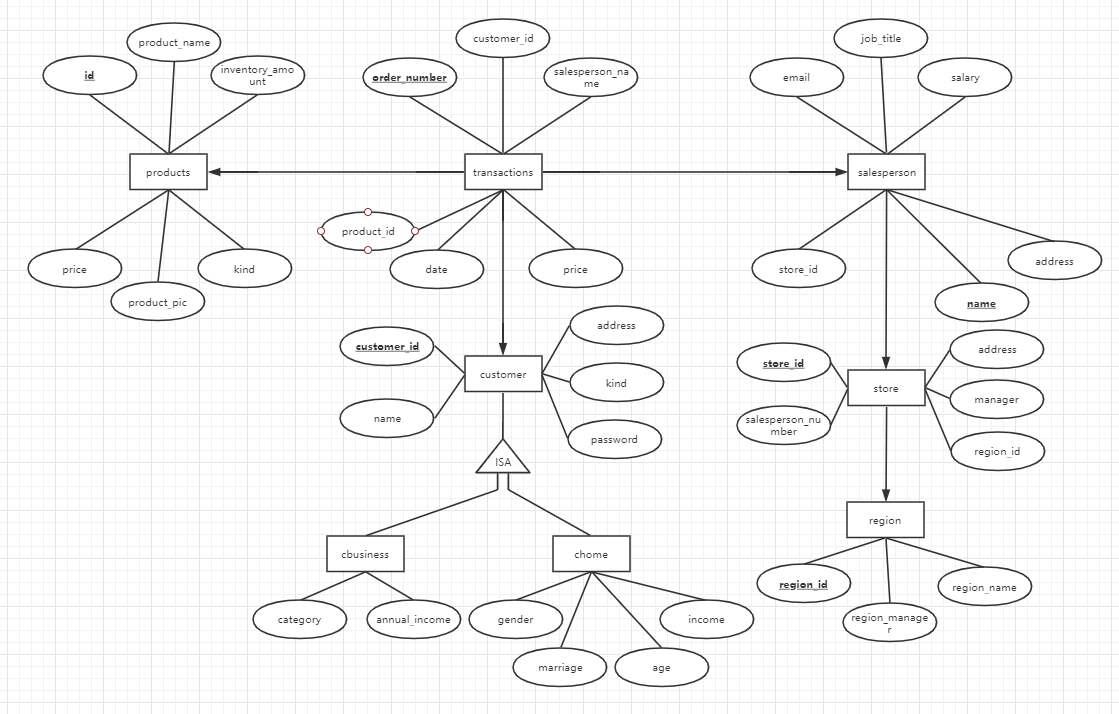
****

Figure 1: ER Diagram

# 4. Database Design

1) DDL statements:

-- phpMyAdmin SQL Dump

-- version 4.8.5

-- https://www.phpmyadmin.net/

--

-- Host: 127.0.0.1:3306

-- Generation Time: Dec 03, 2019 at 07:30 PM

-- Server version: 5.7.26

-- PHP Version: 7.2.18

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET AUTOCOMMIT = 0;

START TRANSACTION;

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- Database: `digital\_shop`

--

-- --------------------------------------------------------

--

-- Table structure for table `cbusiness`

--

DROP TABLE IF EXISTS `cbusiness`;

CREATE TABLE IF NOT EXISTS `cbusiness` (

`customer\_id` varchar(10) NOT NULL,

`category` varchar(20) NOT NULL,

`annual\_income` int(10) NOT NULL,

PRIMARY KEY (`customer\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- --------------------------------------------------------

--

-- Table structure for table `chome`

--

DROP TABLE IF EXISTS `chome`;

CREATE TABLE IF NOT EXISTS `chome` (

`customer\_id` varchar(10) NOT NULL,

`gender` varchar(10) NOT NULL,

`marriage` varchar(10) NOT NULL,

`age` int(3) NOT NULL,

`income` int(10) NOT NULL,

PRIMARY KEY (`customer\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- --------------------------------------------------------

--

-- Table structure for table `customer`

--

DROP TABLE IF EXISTS `customer`;

CREATE TABLE IF NOT EXISTS `customer` (

`customer\_id` varchar(10) NOT NULL,

`name` varchar(40) NOT NULL,

`address` varchar(80) NOT NULL,

`kind` varchar(10) NOT NULL,

`password` varchar(10) NOT NULL,

PRIMARY KEY (`customer\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- --------------------------------------------------------

--

-- Table structure for table `products`

--

DROP TABLE IF EXISTS `products`;

CREATE TABLE IF NOT EXISTS `products` (

`id` varchar(10) NOT NULL,

`product\_name` varchar(40) NOT NULL,

`inventory\_amount` int(10) NOT NULL,

`price` int(10) NOT NULL,

`kind` varchar(20) NOT NULL,

`product\_pic` varchar(40) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- --------------------------------------------------------

--

-- Table structure for table `region`

--

DROP TABLE IF EXISTS `region`;

CREATE TABLE IF NOT EXISTS `region` (

`region\_id` varchar(5) NOT NULL,

`region\_name` varchar(40) NOT NULL,

`region\_manager` varchar(40) NOT NULL,

PRIMARY KEY (`region\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- --------------------------------------------------------

--

-- Table structure for table `salesperson`

--

DROP TABLE IF EXISTS `salesperson`;

CREATE TABLE IF NOT EXISTS `salesperson` (

`name` varchar(40) NOT NULL,

`store\_id` varchar(5) NOT NULL,

`address` varchar(80) NOT NULL,

`job\_title` varchar(20) NOT NULL,

`email` varchar(40) NOT NULL,

`salary` int(10) NOT NULL,

PRIMARY KEY (`name`),

KEY `store\_id` (`store\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- --------------------------------------------------------

--

-- Table structure for table `store`

--

DROP TABLE IF EXISTS `store`;

CREATE TABLE IF NOT EXISTS `store` (

`store\_id` varchar(5) NOT NULL,

`address` varchar(40) NOT NULL,

`salesperson\_number` int(2) NOT NULL,

`manager` varchar(40) NOT NULL,

`region\_id` varchar(5) NOT NULL,

PRIMARY KEY (`store\_id`),

KEY `region\_id` (`region\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- --------------------------------------------------------

--

-- Table structure for table `transactions`

--

DROP TABLE IF EXISTS `transactions`;

CREATE TABLE IF NOT EXISTS `transactions` (

`order\_number` varchar(10) NOT NULL,

`customer\_id` varchar(10) NOT NULL,

`salesperson\_name` varchar(40) NOT NULL,

`date` date NOT NULL,

`product\_id` varchar(10) NOT NULL,

`price` int(10) NOT NULL,

PRIMARY KEY (`order\_number`),

KEY `salesperson\_name` (`salesperson\_name`),

KEY `customer\_id` (`customer\_id`),

KEY `product\_id` (`product\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--

-- Constraints for dumped tables

--

--

-- Constraints for table `cbusiness`

--

ALTER TABLE `cbusiness`

ADD CONSTRAINT `cbusiness\_ibfk\_1` FOREIGN KEY (`customer\_id`) REFERENCES `customer` (`customer\_id`) ON DELETE CASCADE ON UPDATE CASCADE;

--

-- Constraints for table `chome`

--

ALTER TABLE `chome`

ADD CONSTRAINT `chome\_ibfk\_1` FOREIGN KEY (`customer\_id`) REFERENCES `customer` (`customer\_id`) ON DELETE CASCADE ON UPDATE CASCADE;

--

-- Constraints for table `salesperson`

--

ALTER TABLE `salesperson`

ADD CONSTRAINT `salesperson\_ibfk\_1` FOREIGN KEY (`store\_id`) REFERENCES `store` (`store\_id`);

--

-- Constraints for table `store`

--

ALTER TABLE `store`

ADD CONSTRAINT `store\_ibfk\_1` FOREIGN KEY (`region\_id`) REFERENCES `region` (`region\_id`);

--

-- Constraints for table `transactions`

--

ALTER TABLE `transactions`

ADD CONSTRAINT `transactions\_ibfk\_1` FOREIGN KEY (`salesperson\_name`) REFERENCES `salesperson` (`name`),

ADD CONSTRAINT `transactions\_ibfk\_2` FOREIGN KEY (`customer\_id`) REFERENCES `customer` (`customer\_id`),

ADD CONSTRAINT `transactions\_ibfk\_3` FOREIGN KEY (`product\_id`) REFERENCES `products` (`id`);

COMMIT;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

2) The following tables are created for the system:

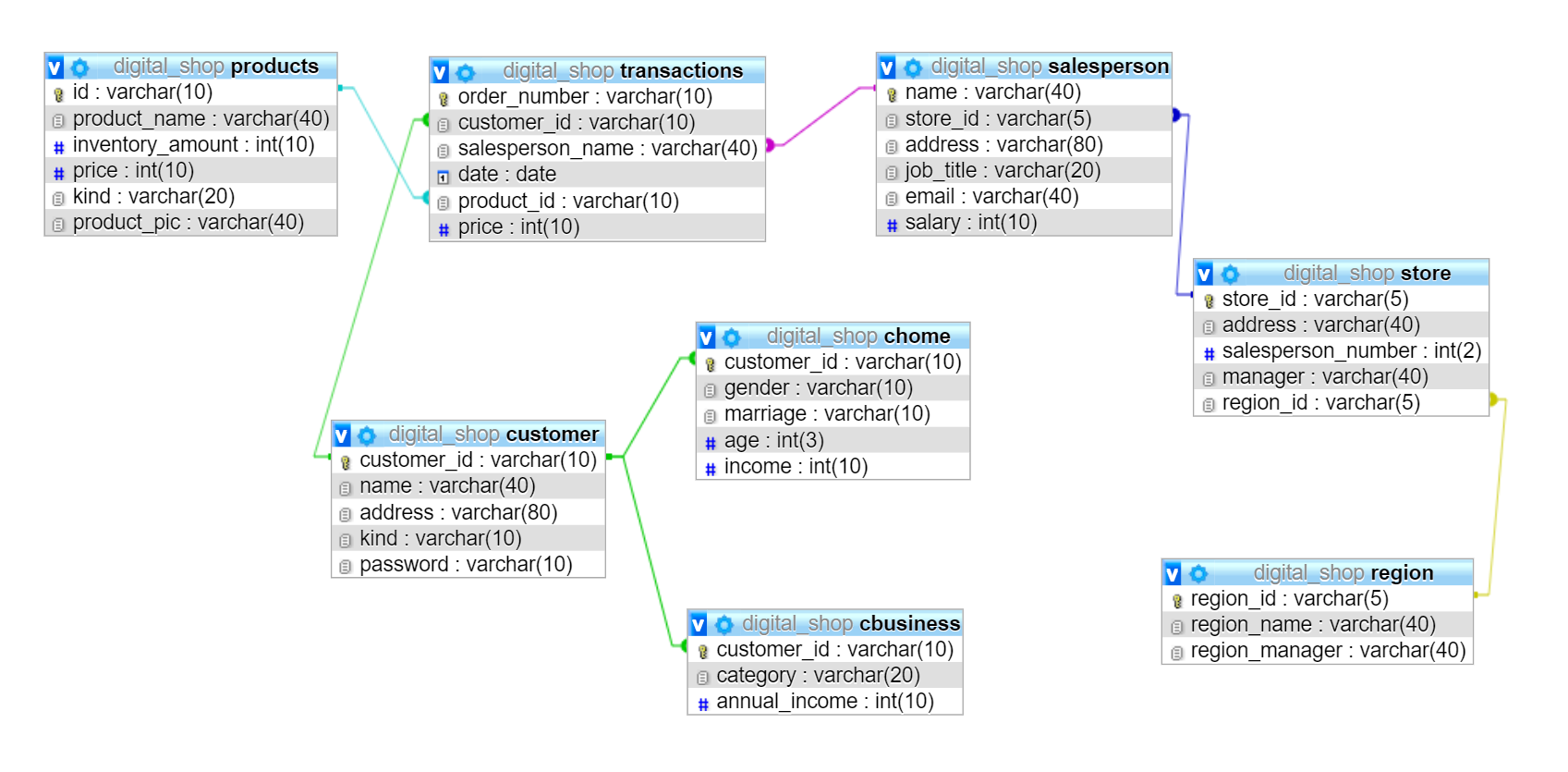
****

Figure 2: Tables in database

# 5. Connection between Front-end and **Back**-end

1) Front-end: HTML, JAVASript

2) Back-end: PHP, mySQL

The application will have a login page for all customers, either home customers or business customers. The customer needs to provide a username and password that match the contents of the database. If the user successfully logs in, his/her id and username will be recorded by the session for the purchase operation. There also will be a role-based registration page, which means users will be asked to choose whether they are a home customer or a business customer, and they must fill in all required information before submitting. Then, the information of the user will be written into table chome or cbusinese.

# 6.  System Implementation

From the home page, customers will be able to browse digital products by certain categories: laptops, phones, cameras, headphones and TV. After click a single product, customer will be able to see the following detail about each digital product: picture, name, amount, price, kind and product pictures. Once the customer buys a product, an order creates. He/She will receive a confirmation ticket for that order along with the pick-up details.

1. Application Screenshots

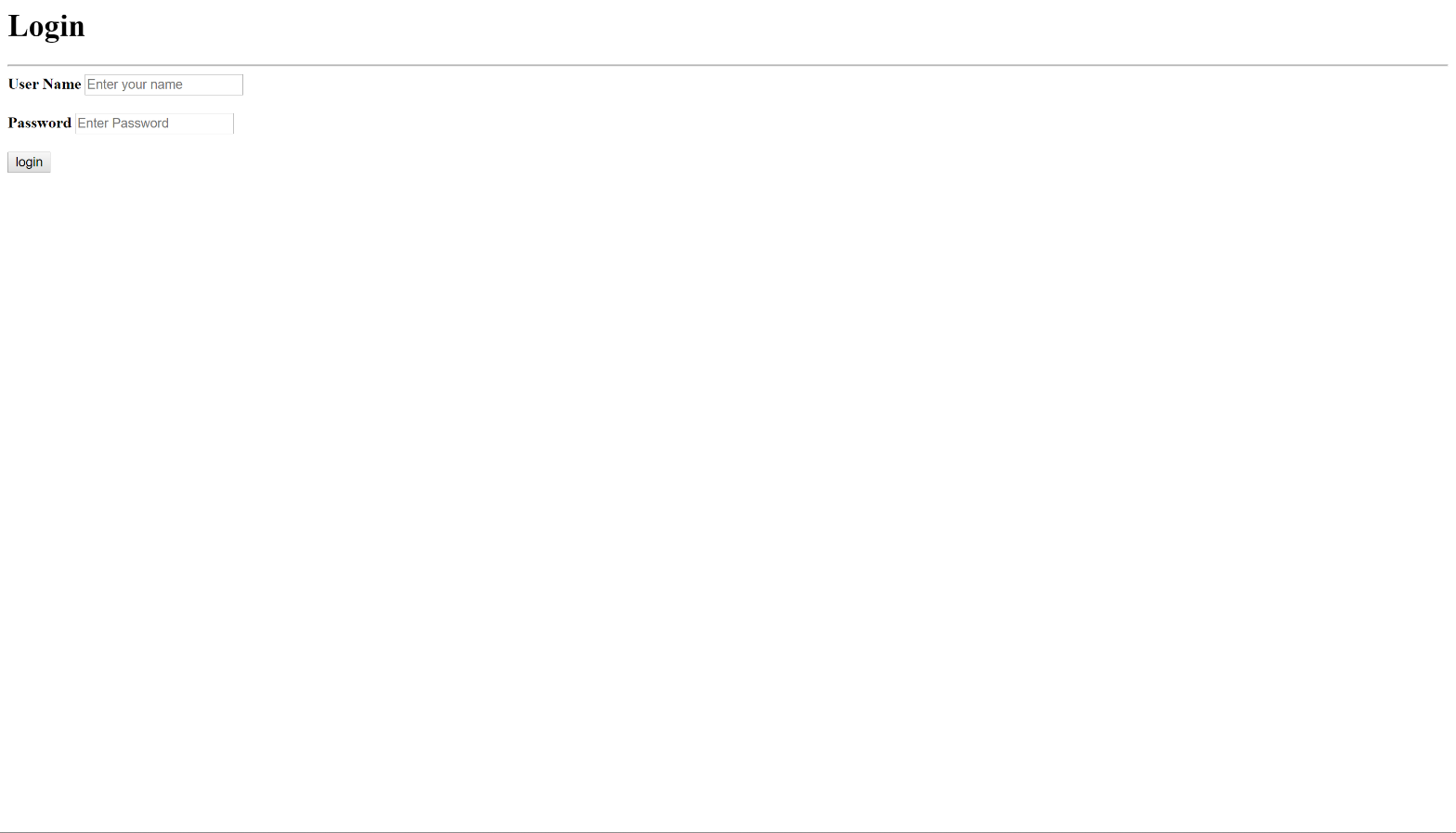
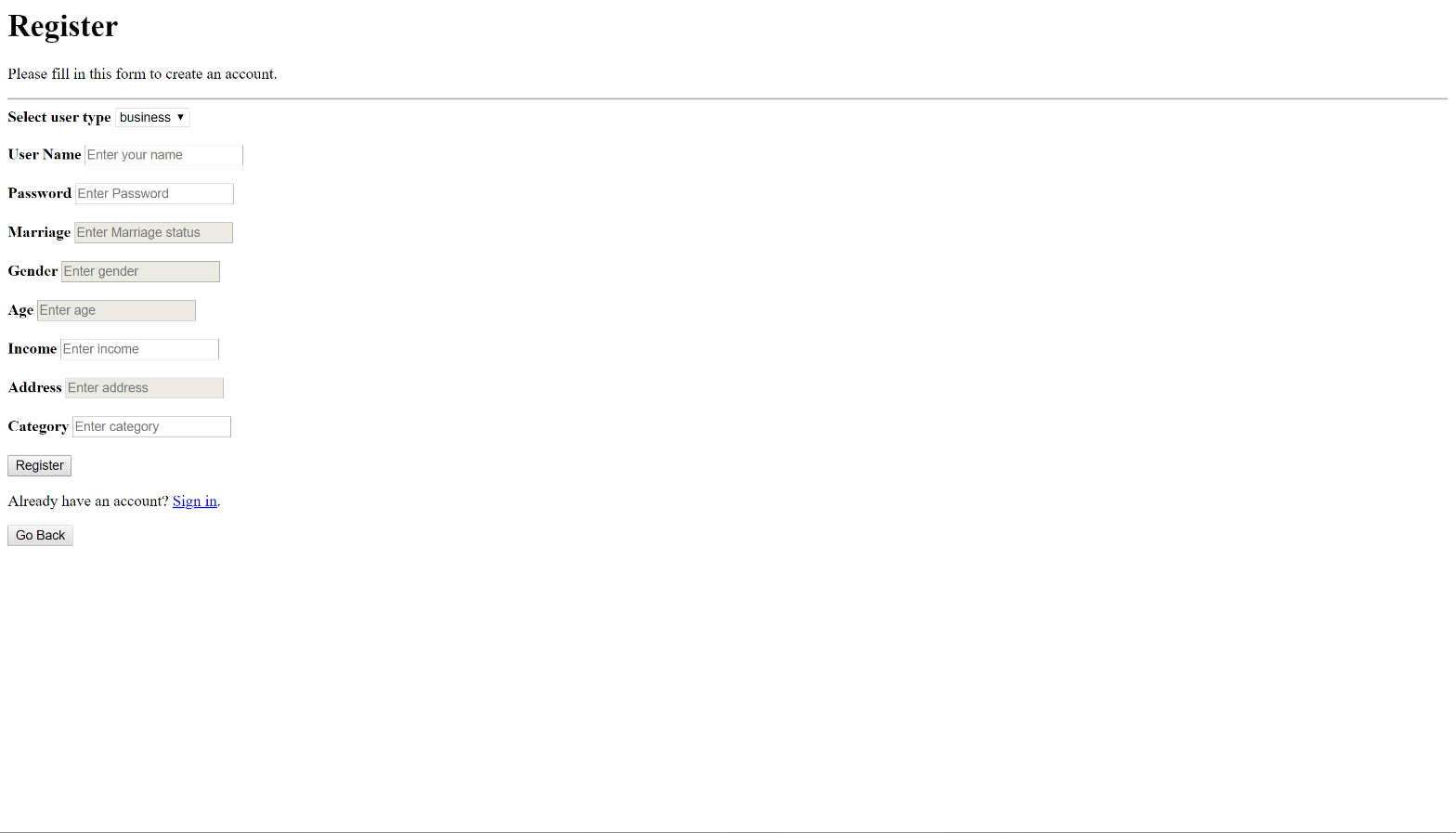
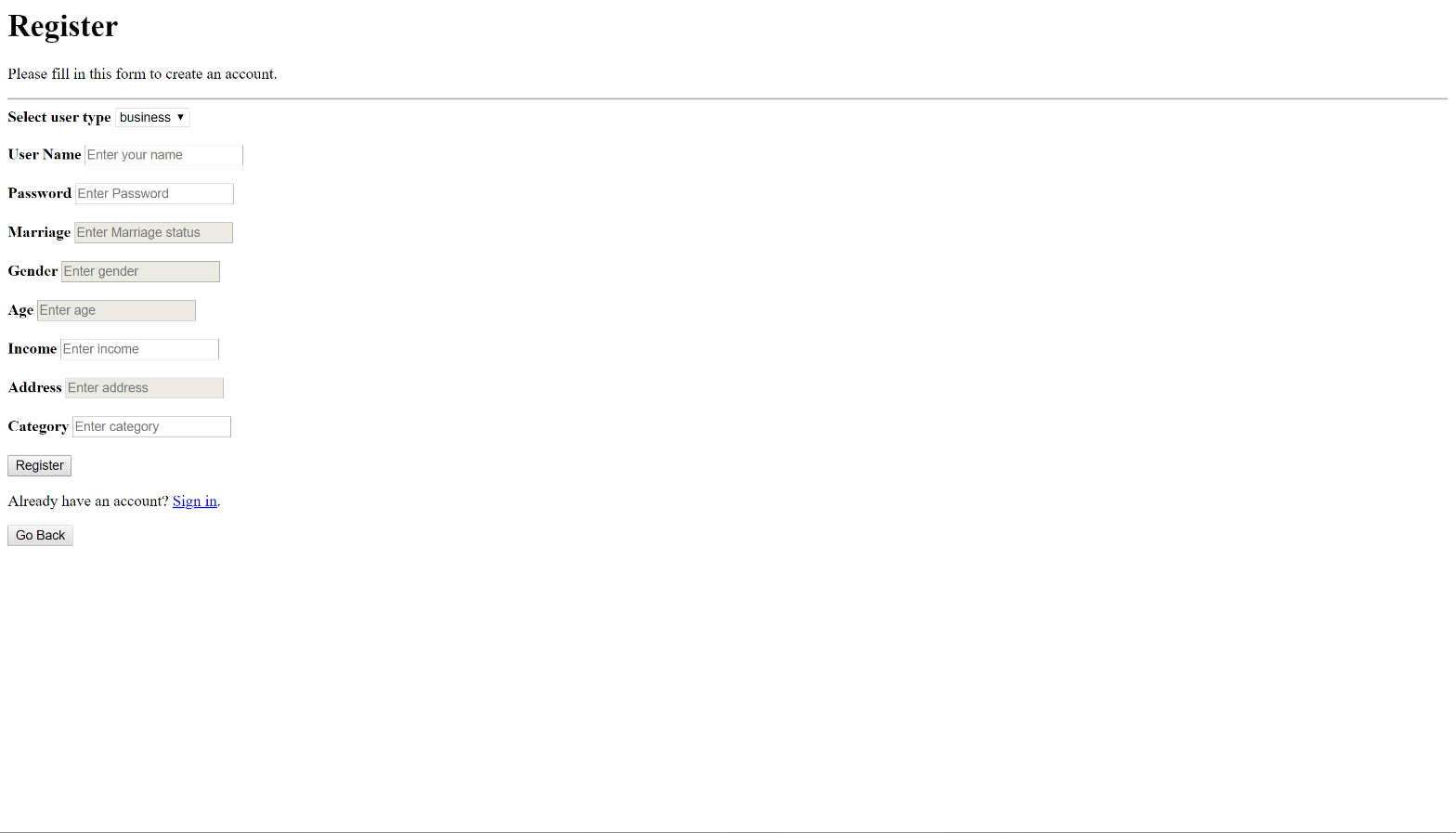


Figure 3: Register and login page

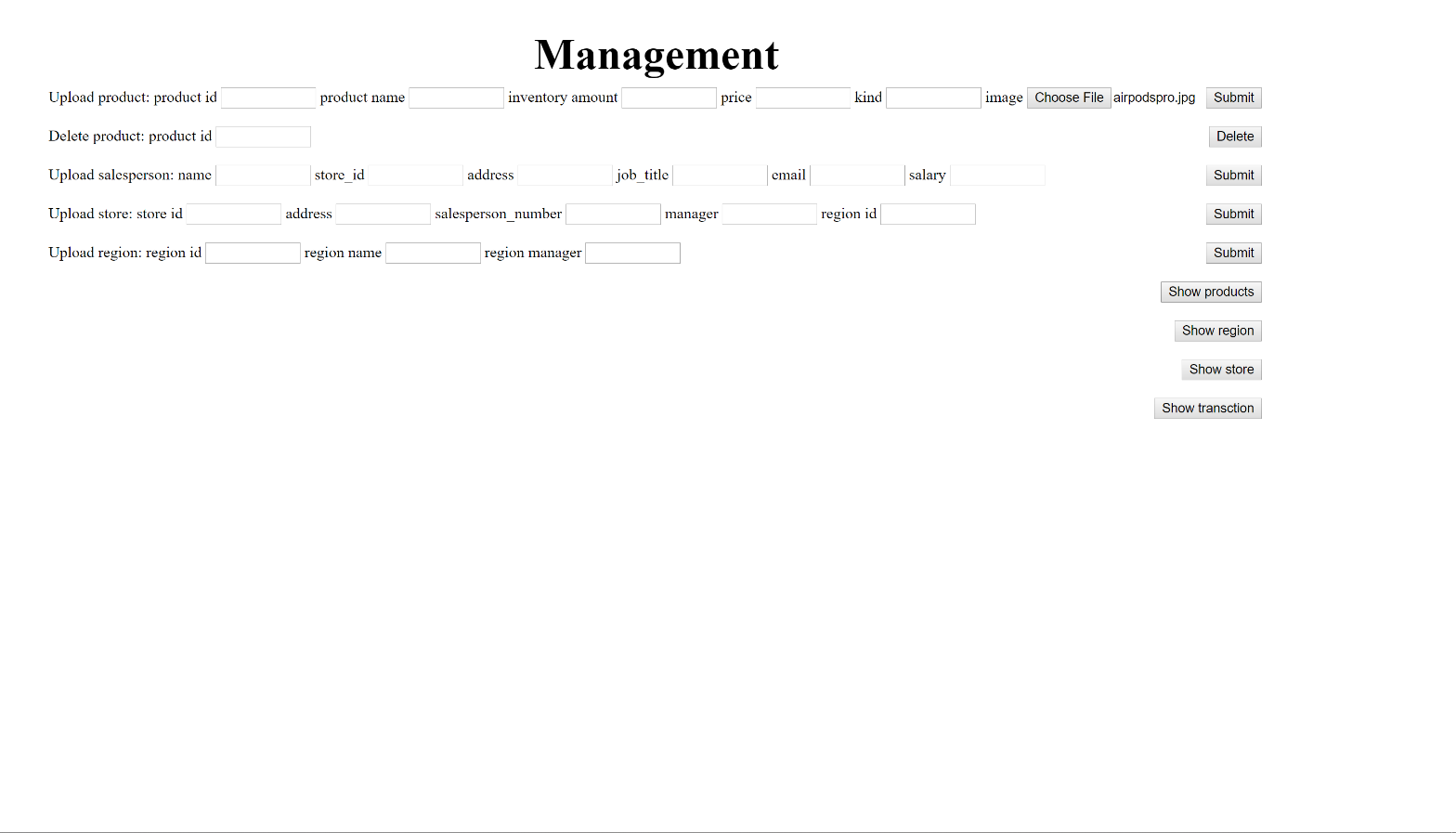


Figure 4: Management page

# 7. Erroneous Cases Handled

After we upload a product image, we cannot load it to the homepage automatically. We tried to modify the php code and found that there was a problem when reading the file path, so we modified the relevant code and finally loaded the image to the corresponding location.

# 8. Limitations and Improvements

1) Limitations

a) We just realized the basic requirements of the project, but the application interface is designed to be very simple and not easy to browse, so the user experience is not good.

b) The function of the application is very simple, and it does not even design a shopping cart function, which is very different from the real online shopping experience.

1. Improvements
2. We can reorganize business logic and improve the interactive interface so that users have a better shopping experience.
3. We can add a shopping cart so that users can modify the shopping data at any time.
4. We can add a data management program to analyze users' purchase records, design recommendation algorithms, and recommend more suitable products for users
5. We can add a logistics management system to facilitate users and merchants to track the progress of express delivery in real time