Final Report

Green life online shopping & management system

University of Pittsburgh

INFSCI 2710

Ziqi Tang

Li Shi

Li Wang

Overview

The “Green life” is an online shopping and items Management System. it is developed for the customers who love shopping online. This system allows the customers to search for the items they want, to purchase the selected items, and to browse the most popular items. Meanwhile, the system also allows the administrators to manage the shopping site and items database. What’s more, the administrators have the highest authority to manage each order and account’s information.

**Modules:**

1. Customer: used for managing the customer details
2. Store: used for managing the store detail
3. Product: store information related to products
4. Salesperson: used to managing the salesperson’s detail, a salesperson is responsible for one or multiple stores
5. Order list: store order history for each customer
6. Address: store customer’s and store’s address

**Functions:**

1. Registration function: Include format check
2. Login/logout function
3. Search function: Include key word search, category search.
4. Purchase goods by placing an order
5. Order list

Assumption:

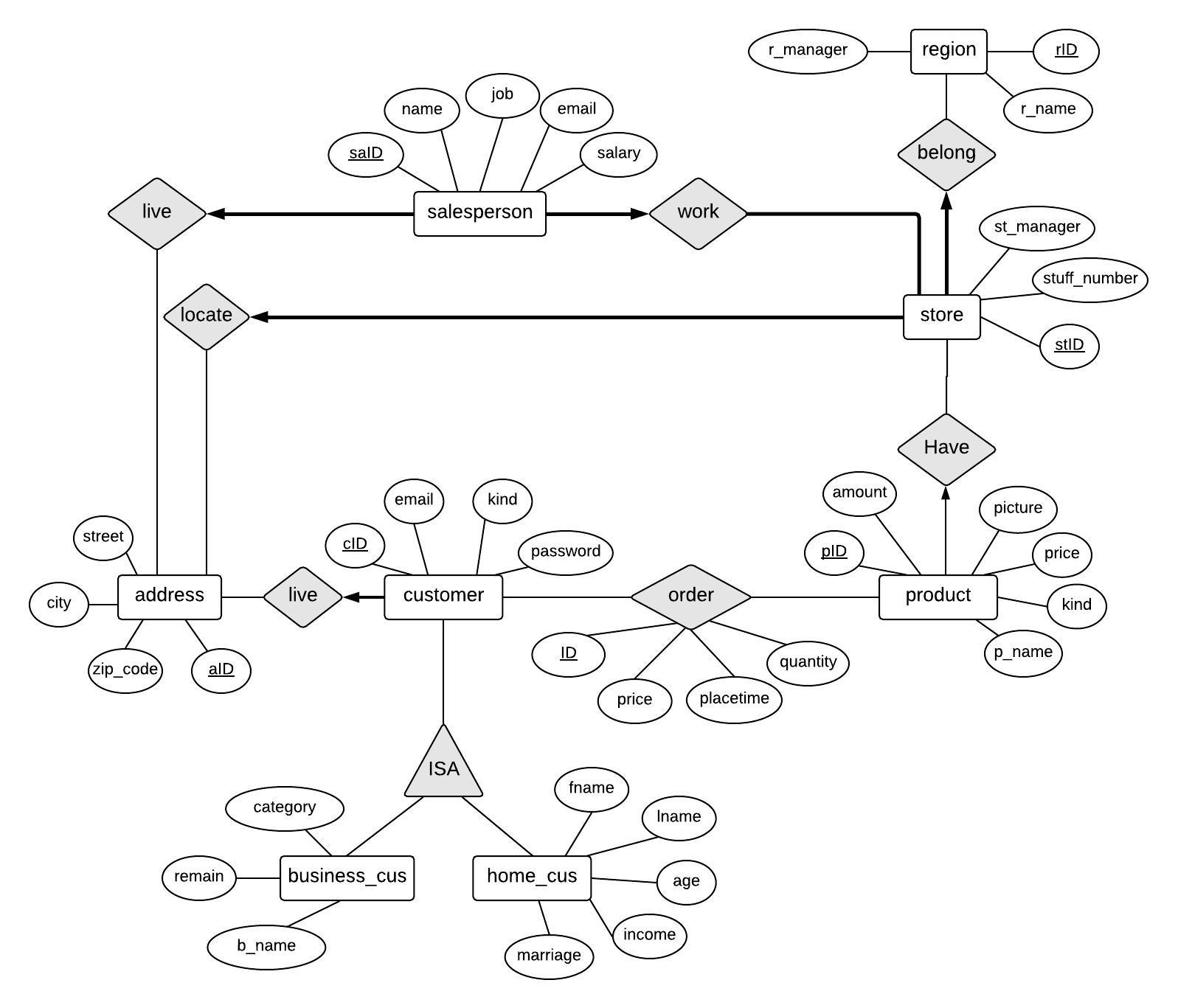
**Customers:**

1. People can use email to create an account, after login, the customer can edit their personal information.
2. Two different kind of customer, one is normal customer, the other is business customer.
3. In the center of the main page, there exists a most popular products view shows the top 3 items in this system.
4. The customers can input key word to search, or use the category button to browse the items.
5. Once the costumers have made their mind, press the “place your order” button, and the system will create the data of this transaction and insert it into database. The customers can check their orders by clicking “My order” button.
6. Add the items to the shopping cart, and then check out with the customer’s payment method.
7. Check the order in the order list page.

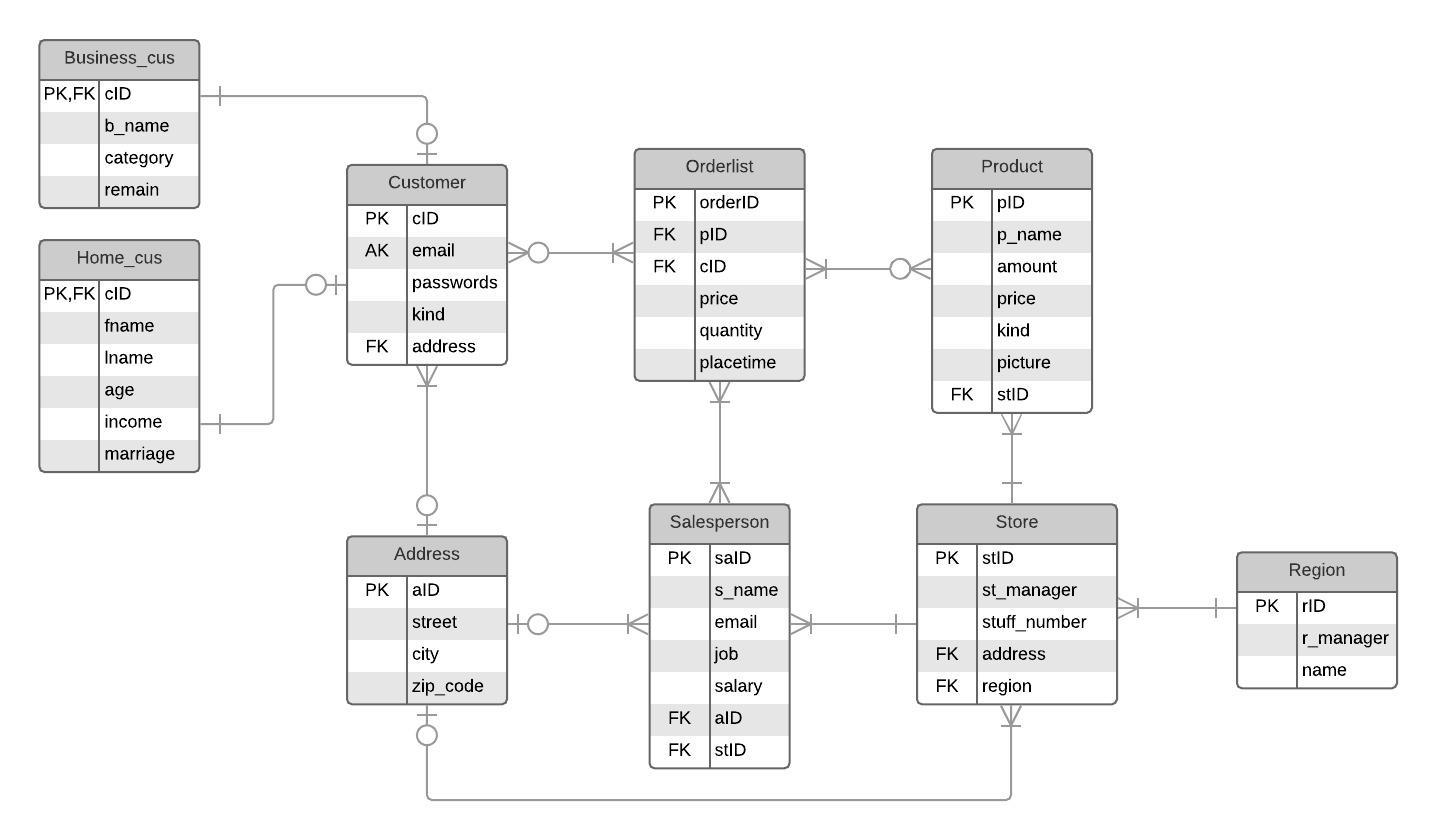
**Administrators**

1. The administrators can use a special page to login and manage the whole system.
2. The administrators have the authority to view and edit customer’s accounts.
3. The administrators can edit the stores information.
4. The administrators can edit the products information.
5. The administrators can view and edit the customers’ transactions.

Graphic Schema



The graph is the E-R diagram of our database. It contains 8 entities and 8 relationships. The entities contain salesperson, store, address, customer, product, business customer and home customer. Since business customer and home customer have same attributes, they have a public entity – customer. The relationship between customer and business customer, home customer is ‘ISA’. The customers live in certain places which are saving in the address entity. One customer must live in one place, but the address ID don’t need to all be used in the relationship with customer (some address tuples save store address and salesperson address). The customer need to order products and one customer can order several different products and the one product can be sold to different customers. The products are in different stores. The store can have several products and the products are in one store. The store is in different regions and one store but located in one region. One region may have several stores. The store also needs to save their location which can be recorded in the address entity. In the stores, there are salespersons working in the stores. One store has several salespersons but one salesperson only can work in one store. Every salesperson has an address.



This is the table structure. We design 9 tables to save the e-commerce database. They are: Business Customer, Home Customer, Customer, Address, Orderlist, Salesperson, Product, Store, and Region. The following is the descriptions of different table attributes.

**Customer:**

1. cID: Auto increase customers’ ID. It’s the primary key of the Customer table.
2. Email: Customers’ email. The email is used as the customers’ log in name so it cannot be duplicated (Alter Key).
3. Passwords: Customers’ password to login the website.
4. Kind: customers is separated into two groups – individual customer and business customer.
5. Address: customers’ address ID. It’s the foreign key which references to the address table.

**Business\_cus**

1. cID: Customers’ ID. It’s the primary key of Business Customer table (Business\_Cus). In the meantime, cID is a reference key which references to the Customer table. So, the business customer can store there data both in customer table and Business\_Cus table.
2. b\_name: name of business company. Since the name of business and customer have different type (Customers’ names include first name and last name), we set different attributes separately in the table.
3. category: kind of business. It means the field that the business is in.
4. remain: the available money in the users’ account. Company and individual usually have different payment method so we set the remain of business and individual separately.

**Home\_cus**

1. cID: Customers’ ID. It’s the primary key of Home Customer table (Home\_Cus). In the meantime, cID is a reference key which references to the Customer table. So, the individual customer can store there data both in customer table and Home\_Cus table.
2. fname: first name of individual customers.
3. lname: last name of individual customers.
4. age: age of individual customers.
5. remain: the available money in the individual customers’ account.
6. marriage: the status of individual customers. ‘1’ represents married and ‘0’ represents single.

**Orderlist**

1. orderID: ID of the orders. It is auto increase attribute and it’s the primary key of order list.
2. pID: the product ID that the customers buy. It’s a foreign key which should reference product table.
3. cID: the customer ID of customer that buy products. It’s a foreign key which should reference customer table.
4. price: the price that customers paid for the order.
5. quantity: the amount of the product that the customer buy. When a order is placed, the amount attribute in the product table should minus the amount that the customer buy.
6. placetime: the time that the order is placed. It is auto put into the order list table.

**Product**

1. pID: ID of products. It is the primary key of product list.
2. p\_name: name of products.
3. amount: the amount of the products that is in store.
4. price: the price of the products.
5. kind: the kind of the products. Include clothes, shoes, beverage, vitamin, cooking, breakfast, baby formula and fresh flowers. The attribute let the users can search the products according to the category.
6. picture: the link of the product picture that need to be shown on the website.
7. stID: the store ID that the product is in. It’s a foreign key which should reference store table.

**Address**

1. aID: ID of address. It is the primary key of address.
2. street: the street address.
3. city: the city address.
4. zip\_code: the address zip\_code.

**Salesperson**

1. saID: ID of salesperson.
2. s\_name: name of salesperson.
3. email: email of salesperson.
4. job: the job title of salesperson.
5. salary: salary of salesperson.
6. aID: address ID of salesperson. It is a foreign key of address table. The salesperson’ address is stored in the address table.
7. stID: the store ID that the salesperson works in. It is a foreign key of store table.

**Store**

1. stID: ID of store.
2. st\_manager: name of the store manager.
3. stuff\_number: number of store stuff.
4. aID: address ID that the store located. It’s a foreign key of address table.
5. rID: region ID that the store is in. It’s a foreign key of region table.
6. Region
7. rID: ID of region.
8. r\_manager: name of region manager.
9. name: the name of the region.

DDL statements:

CREATE TABLE `address` (

`aID` int(11) NOT NULL AUTO\_INCREMENT,

`street` varchar(80) NOT NULL,

`city` varchar(20) NOT NULL,

`zip\_code` varchar(10) NOT NULL,

PRIMARY KEY (`aID`)

) ENGINE=InnoDB AUTO\_INCREMENT=50 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

CREATE TABLE `customer` (

`cID` int(11) NOT NULL AUTO\_INCREMENT,

`email` varchar(30) NOT NULL,

`passwords` varchar(20) NOT NULL,

`kind` varchar(20) NOT NULL,

`aID` int(11) NOT NULL,

PRIMARY KEY (`cID`),

UNIQUE KEY `email` (`email`),

KEY `aID` (`aID`),

CONSTRAINT `customer\_ibfk\_1` FOREIGN KEY (`aID`) REFERENCES `address` (`aid`)

) ENGINE=InnoDB AUTO\_INCREMENT=40 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

CREATE TABLE `business\_cus` (

`cID` int(11) NOT NULL,

`b\_name` varchar(20) NOT NULL,

`remain` decimal(18,2) DEFAULT NULL,

`category` varchar(20) DEFAULT NULL,

PRIMARY KEY (`cID`),

CONSTRAINT `business\_cus\_ibfk\_1` FOREIGN KEY (`cID`) REFERENCES `customer` (`cid`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

CREATE TABLE `home\_cus` (

`cID` int(11) NOT NULL,

`fname` varchar(20) NOT NULL,

`lname` varchar(20) NOT NULL,

`age` varchar(10) DEFAULT NULL,

`marriage` int(11) DEFAULT NULL,

`remain` decimal(18,2) NOT NULL,

PRIMARY KEY (`cID`),

CONSTRAINT `home\_cus\_ibfk\_1` FOREIGN KEY (`cID`) REFERENCES `customer` (`cid`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

CREATE TABLE `orderList` (

`ID` int(11) NOT NULL AUTO\_INCREMENT,

`cID` int(11) NOT NULL,

`pID` int(11) NOT NULL,

`quantity` int(11) NOT NULL,

`price` decimal(18,2) DEFAULT NULL,

`placetime` datetime DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (`ID`),

KEY `pID` (`pID`),

KEY `cID` (`cID`),

CONSTRAINT `orderlist\_ibfk\_1` FOREIGN KEY (`pID`) REFERENCES `product` (`pid`),

CONSTRAINT `orderlist\_ibfk\_2` FOREIGN KEY (`cID`) REFERENCES `customer` (`cid`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

CREATE TABLE `product` (

`pID` int(11) NOT NULL AUTO\_INCREMENT,

`p\_name` varchar(150) DEFAULT NULL,

`amount` int(11) NOT NULL,

`price` decimal(18,2) NOT NULL,

`kind` varchar(20) NOT NULL,

`picture` varchar(150) DEFAULT NULL,

`stID` int(11) DEFAULT NULL,

PRIMARY KEY (`pID`),

KEY `fk\_store` (`stID`),

CONSTRAINT `fk\_store` FOREIGN KEY (`stID`) REFERENCES `store` (`stid`) ON DELETE SET NULL ON UPDATE SET NULL

) ENGINE=InnoDB AUTO\_INCREMENT=319 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

CREATE TABLE `region` (

`rID` int(11) NOT NULL AUTO\_INCREMENT,

`r\_manager` varchar(20) NOT NULL,

`r\_name` varchar(20) NOT NULL,

PRIMARY KEY (`rID`)

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

CREATE TABLE `salesperson` (

`saID` int(11) NOT NULL AUTO\_INCREMENT,

`s\_name` varchar(20) DEFAULT NULL,

`email` varchar(30) DEFAULT NULL,

`job` varchar(30) DEFAULT NULL,

`salary` decimal(18,2) DEFAULT NULL,

`aID` int(11) DEFAULT NULL,

`stID` int(11) DEFAULT NULL,

PRIMARY KEY (`saID`),

KEY `aID` (`aID`),

KEY `stID` (`stID`),

CONSTRAINT `salesperson\_ibfk\_1` FOREIGN KEY (`aID`) REFERENCES `address` (`aid`) ON DELETE CASCADE,

CONSTRAINT `salesperson\_ibfk\_2` FOREIGN KEY (`stID`) REFERENCES `store` (`stid`) ON DELETE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

CREATE TABLE `store` (

`stID` int(11) NOT NULL AUTO\_INCREMENT,

`st\_manager` varchar(20) DEFAULT NULL,

`stuff\_number` varchar(10) DEFAULT NULL,

`aID` int(11) NOT NULL,

`rID` int(11) NOT NULL,

PRIMARY KEY (`stID`),

KEY `aID` (`aID`),

KEY `rID` (`rID`),

CONSTRAINT `store\_ibfk\_1` FOREIGN KEY (`aID`) REFERENCES `address` (`aid`) ON DELETE CASCADE,

CONSTRAINT `store\_ibfk\_2` FOREIGN KEY (`rID`) REFERENCES `region` (`rid`) ON DELETE CASCADE

)

ENGINE=InnoDB AUTO\_INCREMENT=18 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

Description of Front-end & Back-end:

**Front-end design:**

We used HTML, CSS(Bootstrap) and JQuery to design our website and its function, and then we used AJAX to transfer data.

For the main page of the system, it allows a visitor to visit the website, browse the content , search and buy items. Since the online shop is “GreenLife”, we use green as theme color.

1. Login/logout

Before login, user cannot see the “Log out” button, “My order” button. After having logged in, the login button will disappear. We did not make login module a separate page. It is a bootstrap model, which can show on the top of the page, and disappear by clicking area outside the login area.

The logout is linked to homepage.

1. Register

Personal user or company user use the same register page. The form automatically changes based on which type of user the user chooses. Each form item is data type restricted. For example, if it is a column for age, the type will be restricted to “number”.

1. Search

Search button can detect illegal input, like null, “”, etc. It is right below the navigation, since people will do intensively searching operation.

1. Category

It is on the left side, displaying all the kinds of goods we have.

1. Display the goods

We have pagination to display the items in case the .html page will be too long. Each page contains 4 item to show. User can navigate to the first and last page. There is an “add” button for user to put the item to the cart. The most popular items are displayed in a bootstrap carousel.

1. Shopping cart button

The shopping cart is fully automatic. You can add an item or remove it if you do not want to buy later. The number of total kinds of goods you will purchase will be automatically displayed. The default quantity to buy is 1.

1. Order history

In Order history, the history orders of current user are listed in a table, neatly.

**Back-end design:**

We use python’s SqlAlchemy to manipulate data in Mysql. The back-end process the request from front-end by Flask.

1. Response to homepage:

Back-end should process Get request and Post request. The Get request includes display commodity and the 3 most popular items people have ordered. It covers the **search** and **aggregation** of the data. Also, it processes user’s login information. Back-end will send json-type data to tell user if the login is successful or not. If it succeeds, server will maintain a session for the user. The session will expire once the user closes the browser.

1. Search:

User can search by inputting some key words. If he wants to emphasize on particular addresses, he can type “in (street name)”. Server will treat content after the last “in” as a particular street. The displaying products by its categories, showing order history, displaying most popular products are safe searching operations too.

1. Register:

There are two kinds of register methods, for personal user and company user. We make them in one **transaction**. It includes “**insert”** operation on multiple tables. If error occurs, the transaction will roll back to the initial state where no table is written.

1. Place an order:

This includes search based on multiple table. We have considered the storage change on product, balance change on customer, etc. We have avoided illegal transaction, like buy more products than remaining quantity or over balance behavior. It is a transaction, too. We protect the ACID feature of database.

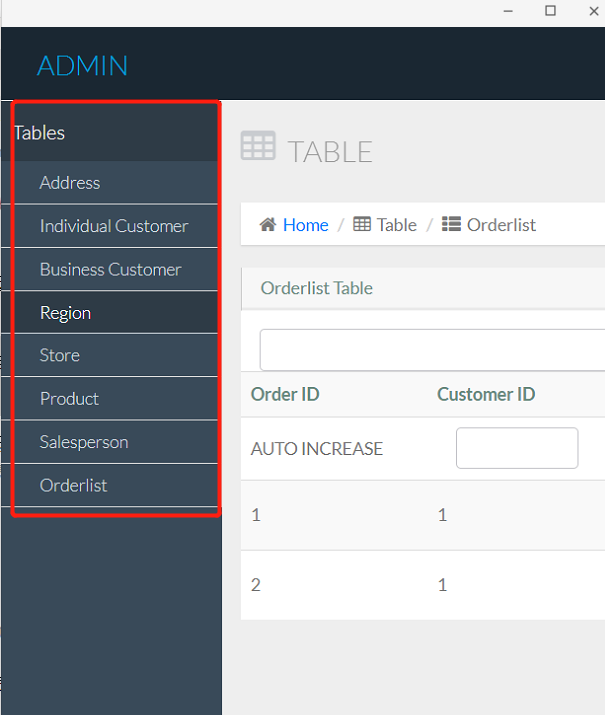
1. Error detection:

We detect many errors, like illegal input, damage to database integrity, etc. And we will transfer the error message to user by render the .html page or send response containing message.

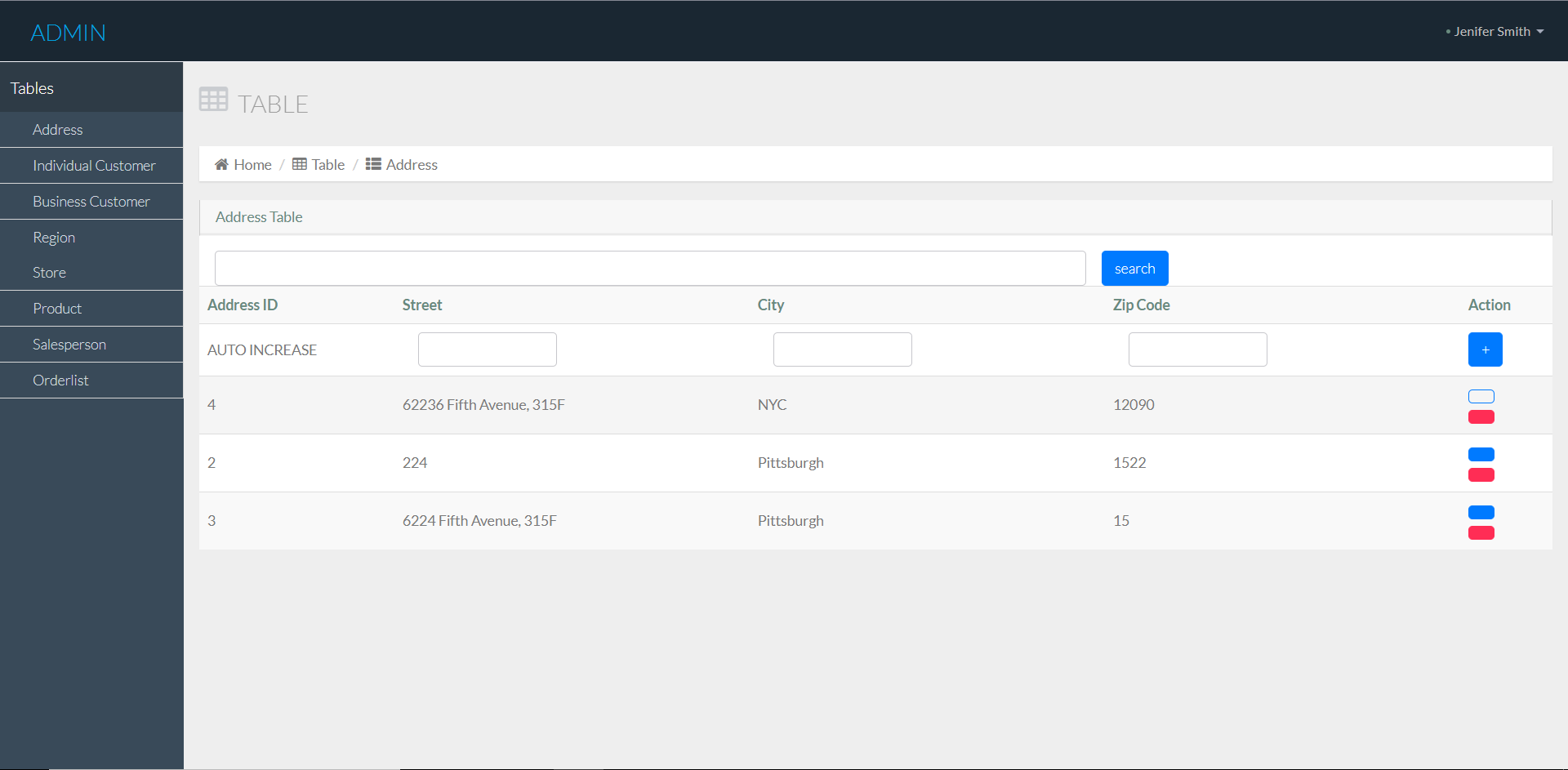
System Implementation:

**Administrator：**

1. The administrator can use this page to edit the data in the database.

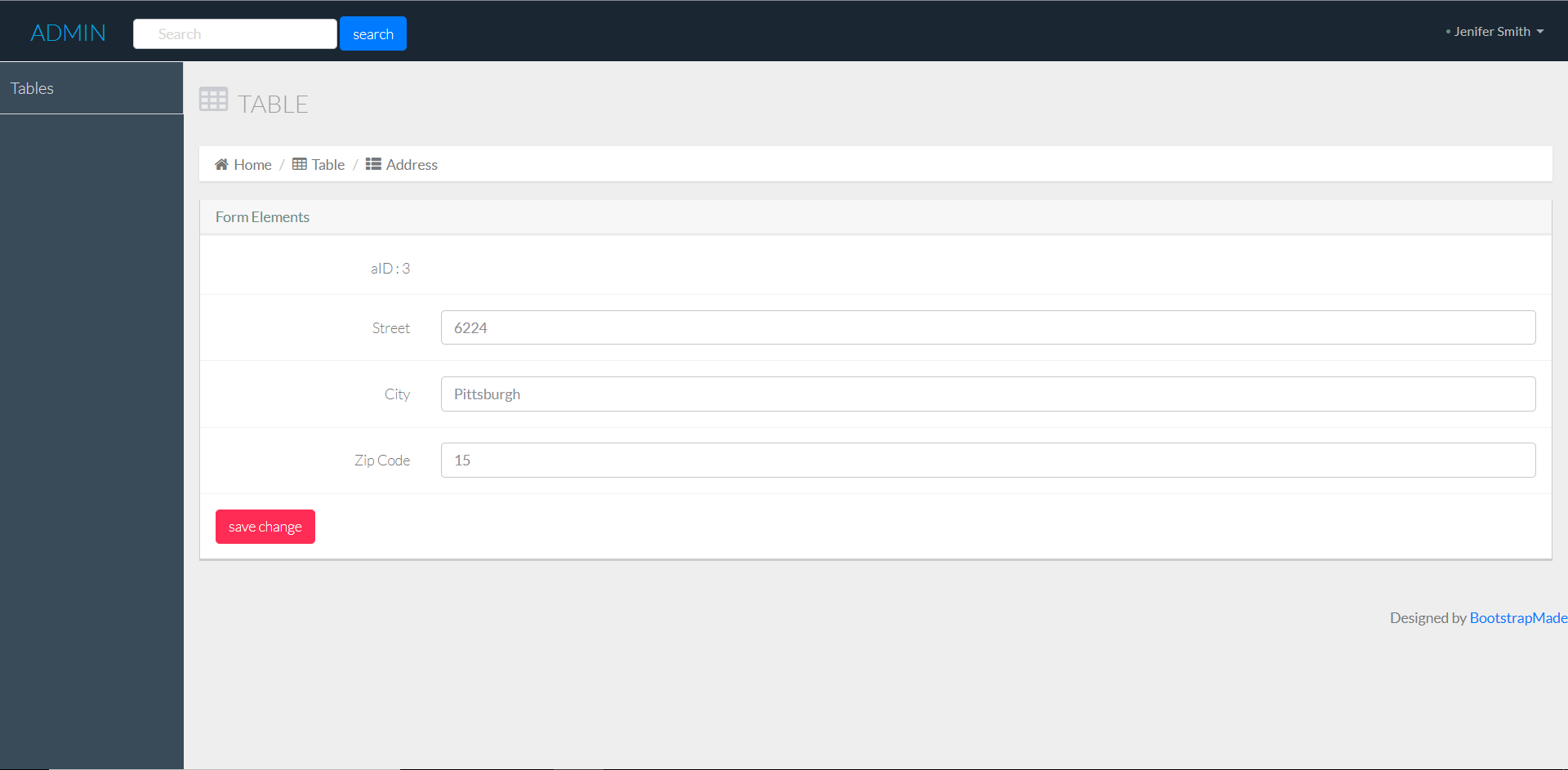


The administrators can change every table in the database. The tables are shown as a list of button above. After clicking the button, the administrator will get access to the corresponding table-update-page.



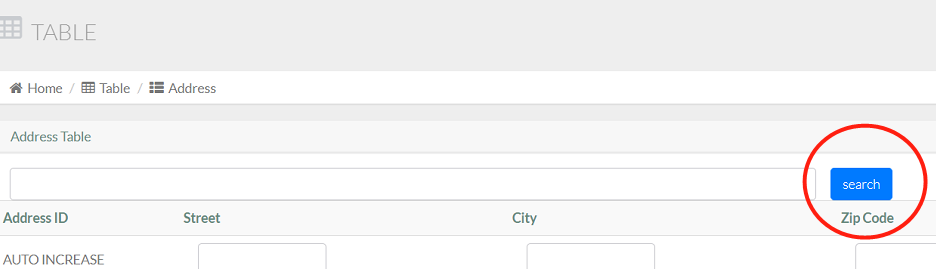
When the administrators press the blue button, they can delete the corresponding column. When press the red button, they can edit the data, and update to the database.

1. After clicking the red button, the administrator can use this page to update the data.

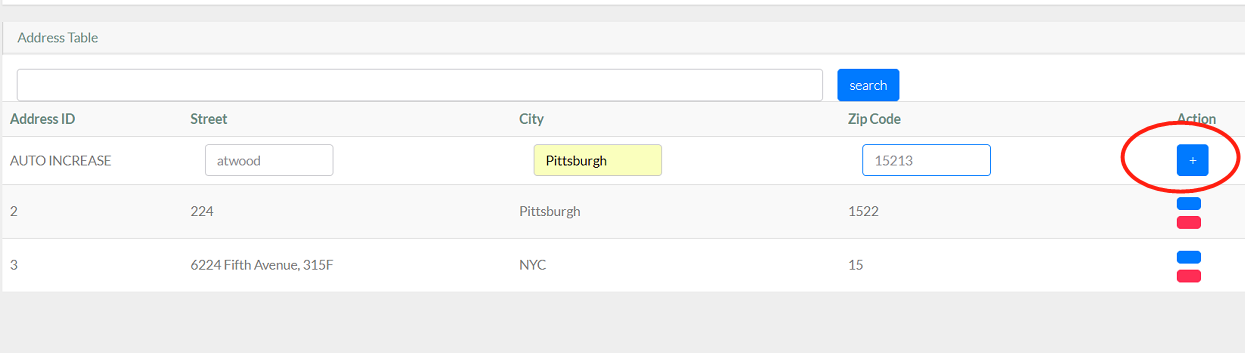


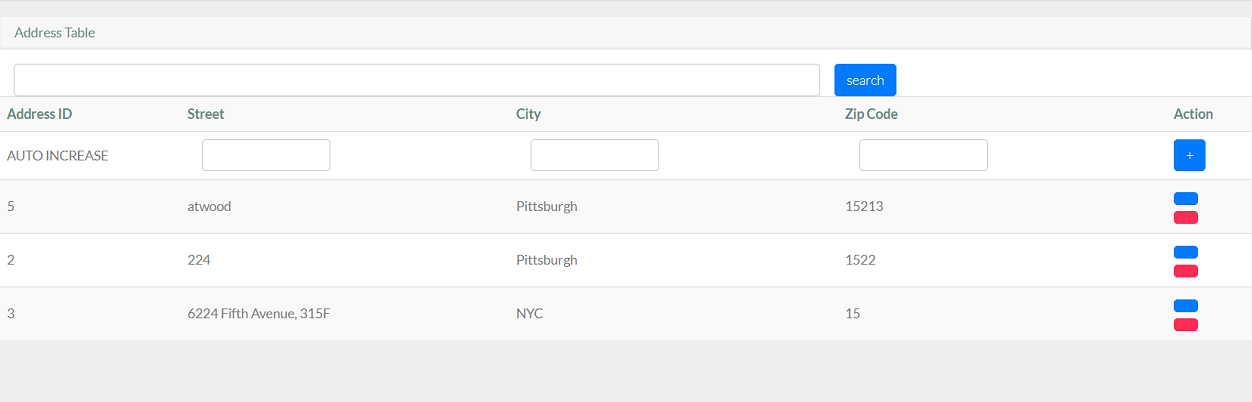
After finishing changes, click “save change” button to commit the change to the database.

1. After inputting the information, press “search” button to search the database.



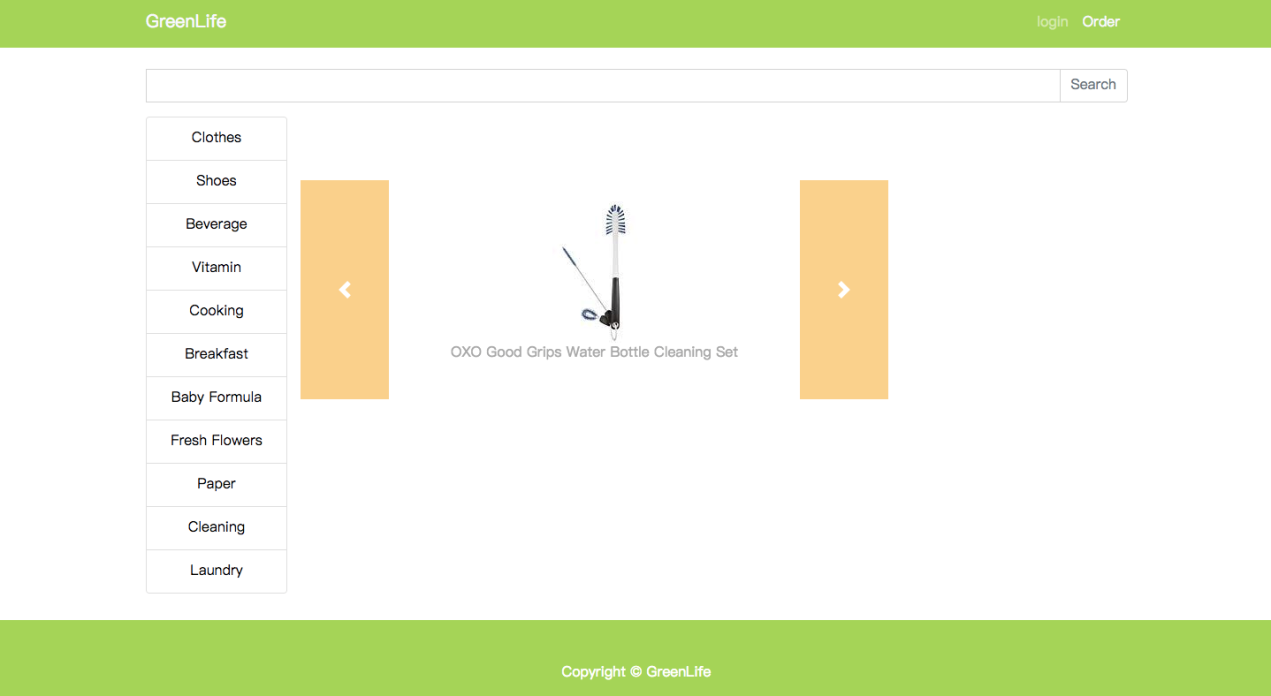
1. By pressing the “+” button, the administrator can add data to the database.





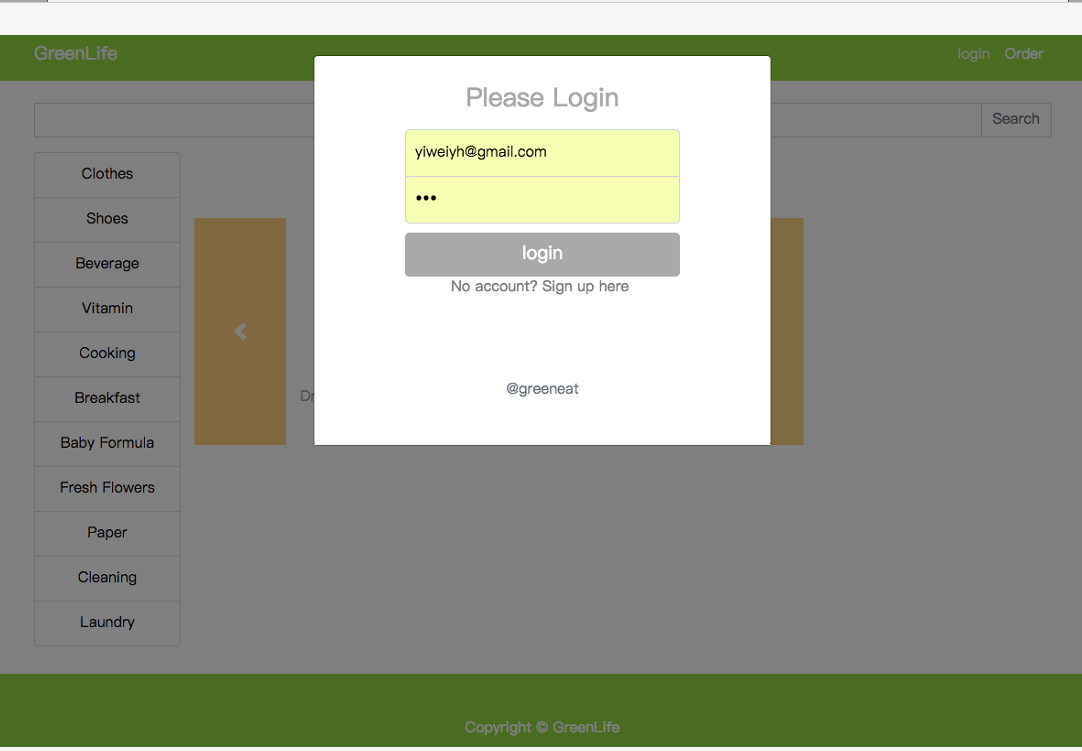
**Customer：**

1. Main page



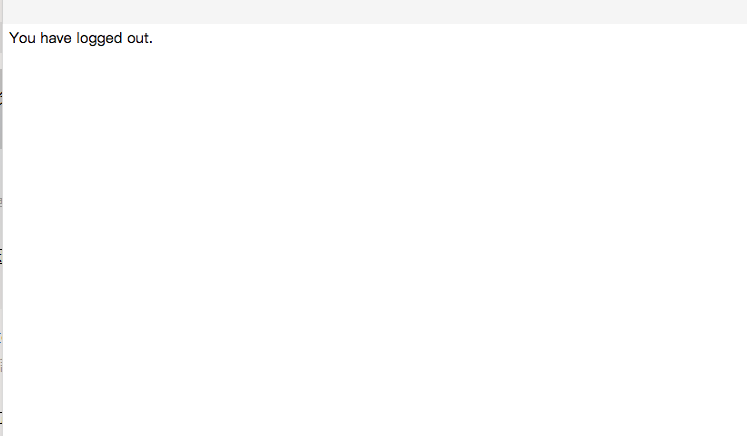
The main page include search by category, search by keywords (include product’s name and address), and display the top 3 popular products.

1. Login

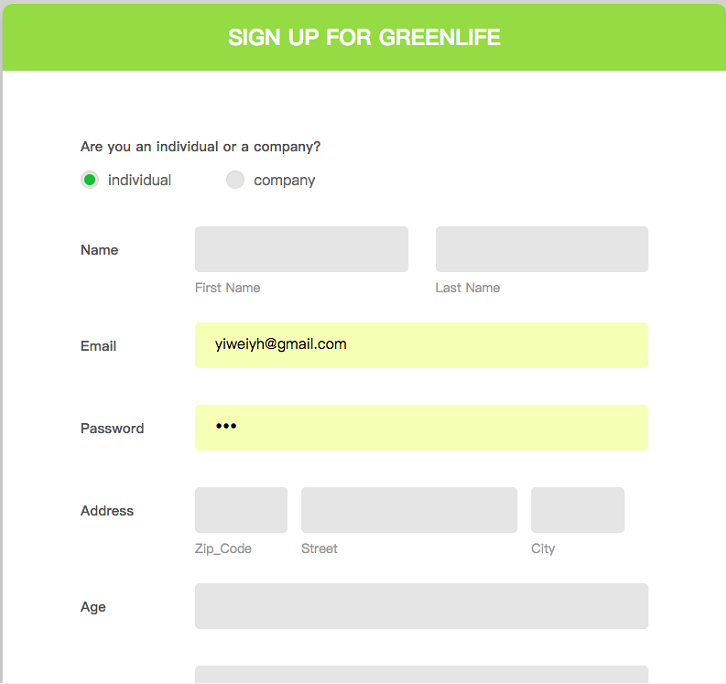


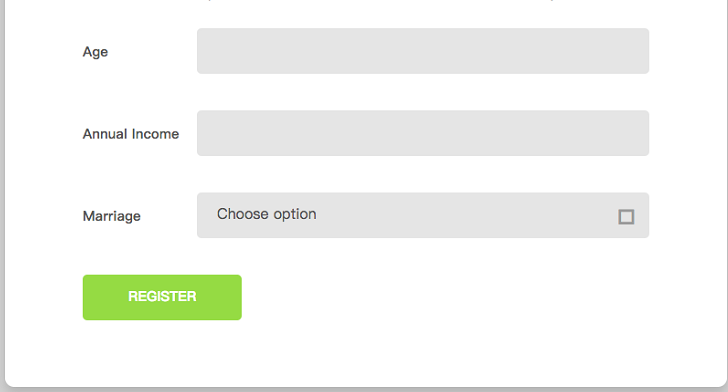
If the visitor don’t have an account, the can press “No account? Sign up here” to get access to the registration page.

1. Logout

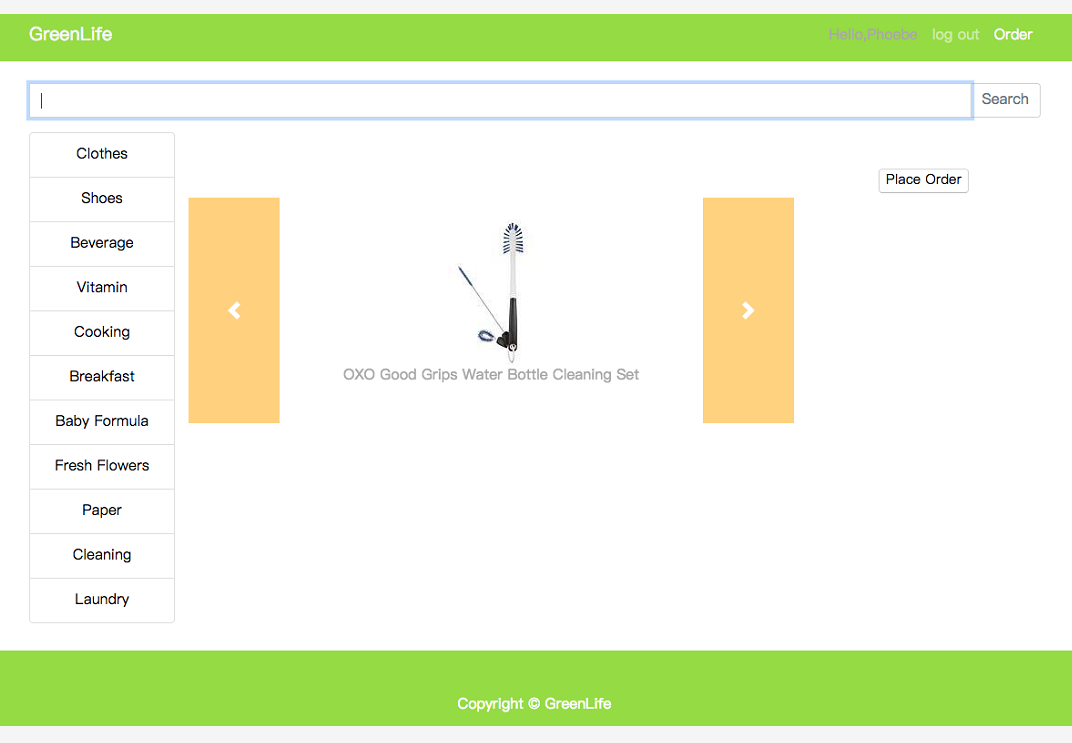


1. Registration page



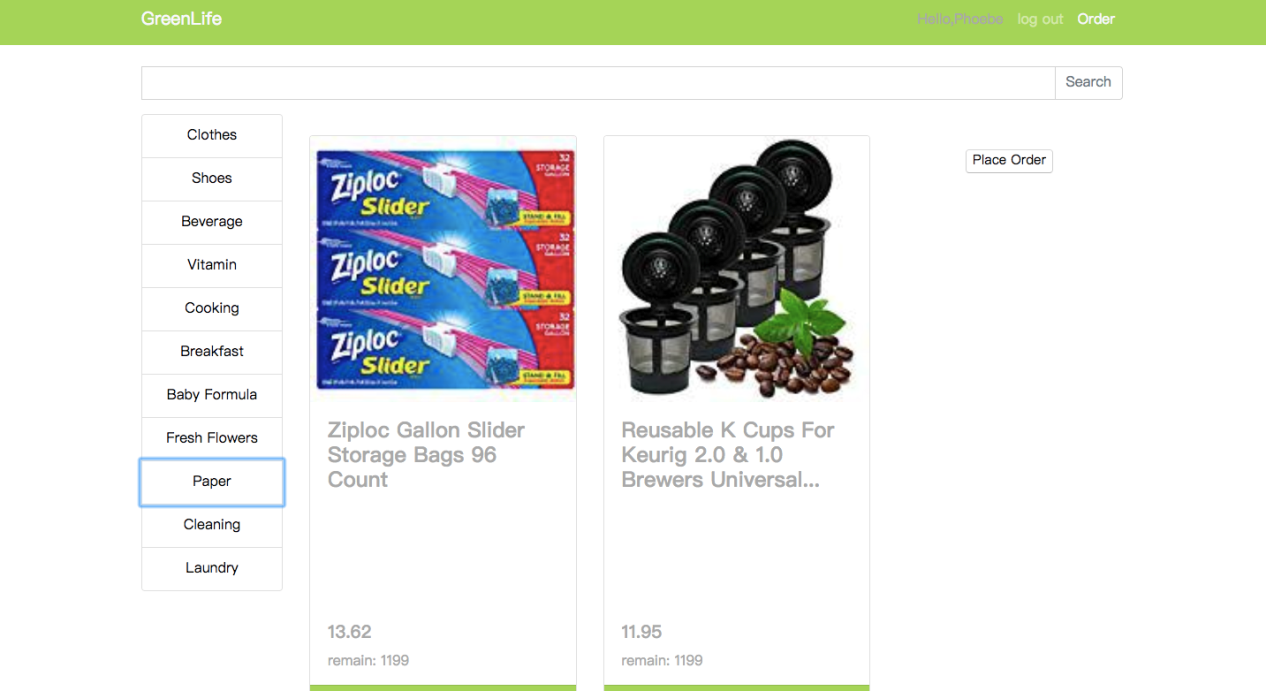


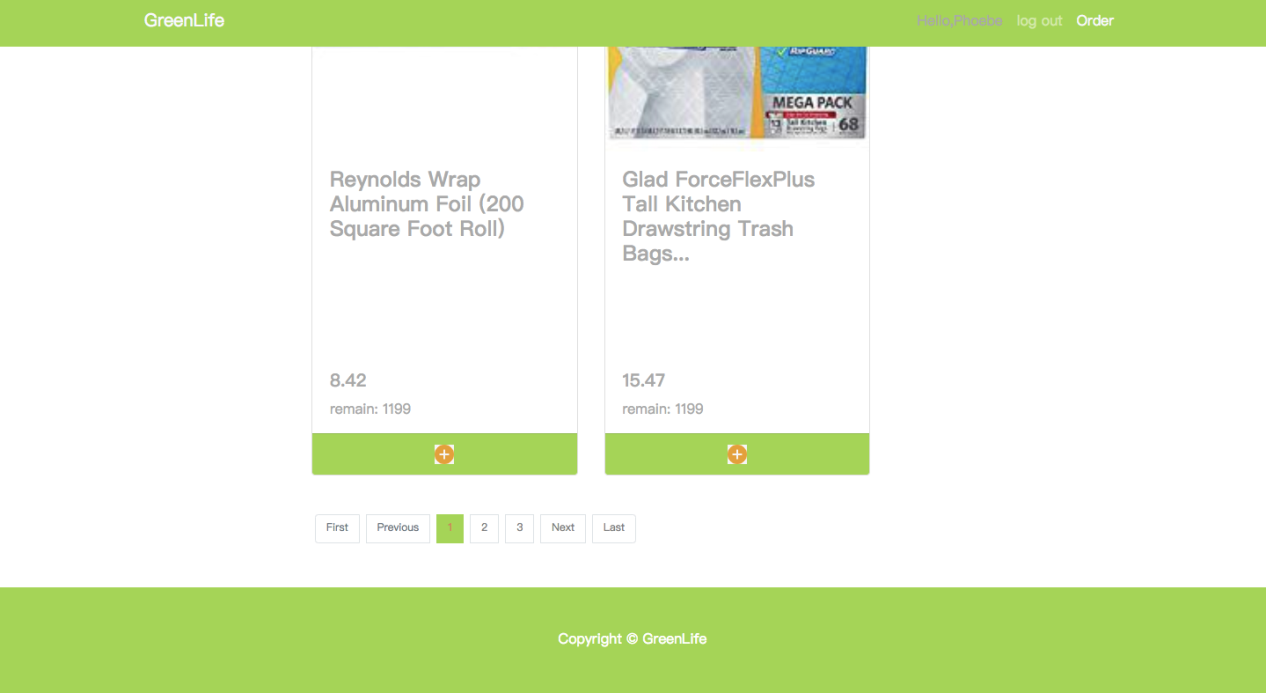
If the information entered is in a wrong format, a corresponding prompt will pop up.

1. Page after login 

After login, the page will display “Hello, XXX” and the customers can choose to logout, they can also press “Place Order” to purchase.

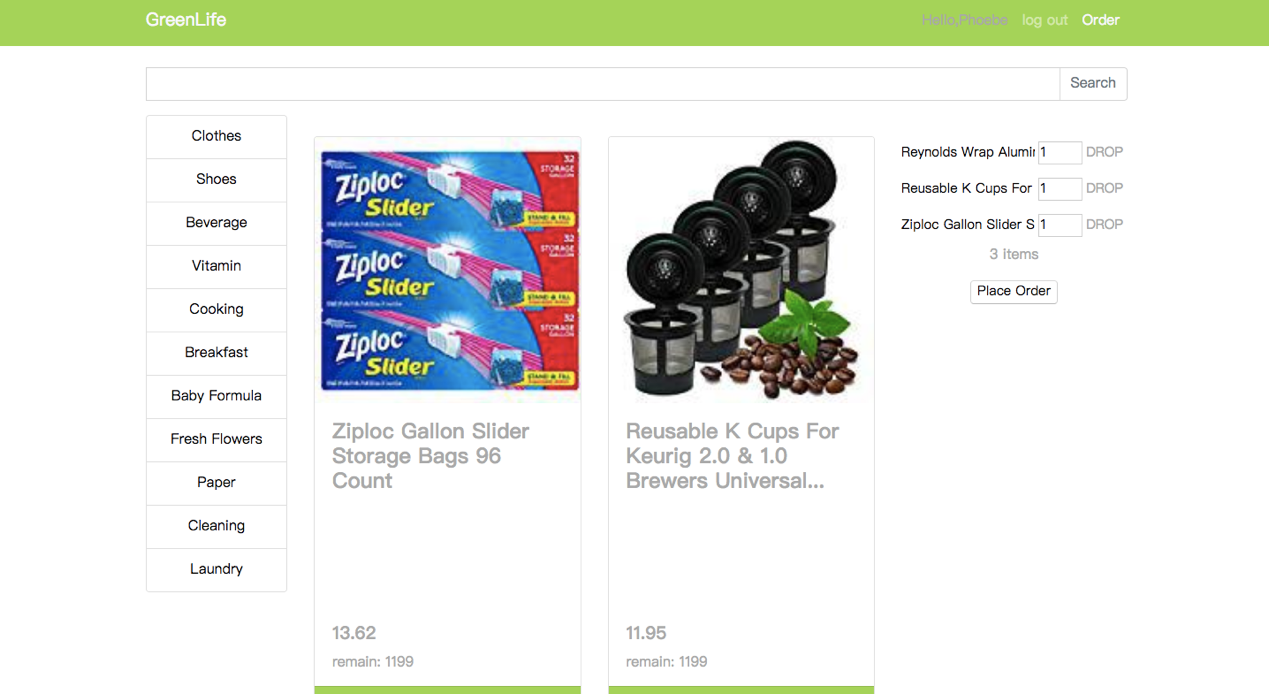
1. Items display





This page display the items’ information (include name, price amount), customer can add the items to the shopping cart. If the number of the items is more than 4, the customer can find a page view at the end of the page and some button to change the page.

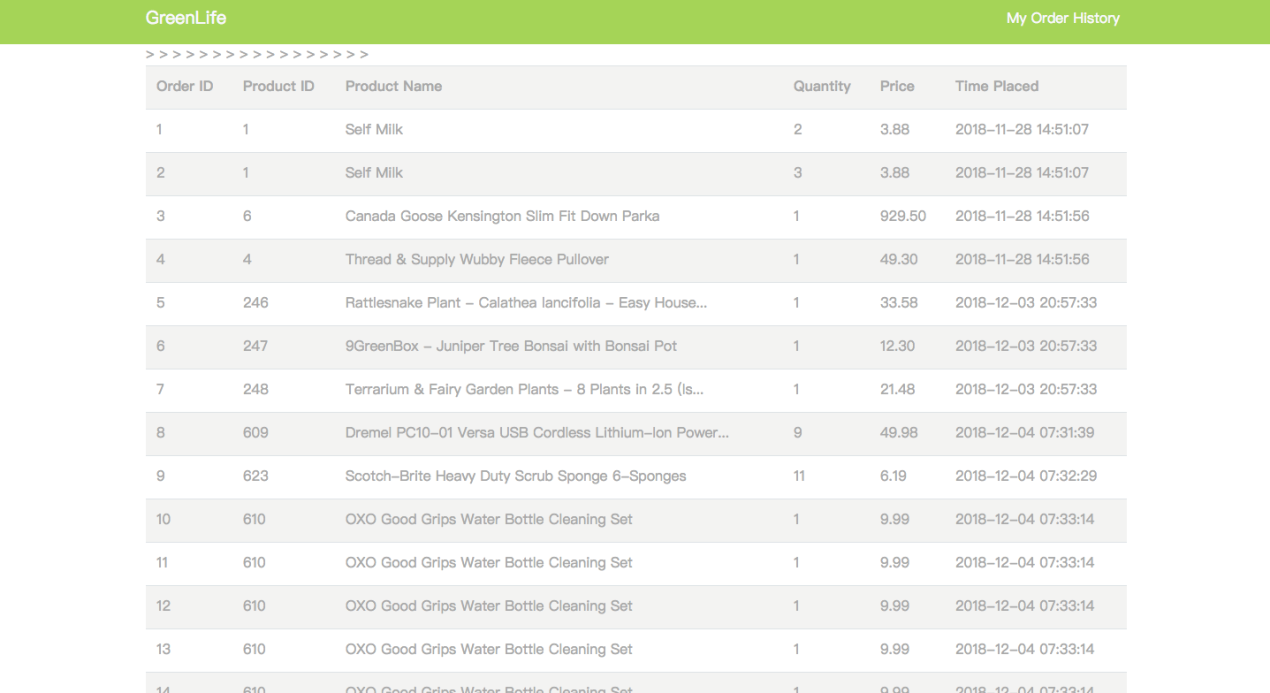
1. Shopping cart



The default number of items is “1”, and there is a “Place Order” button to finish purchase.

1. Order history page

After click the order button in the main page, the system shows the order page.

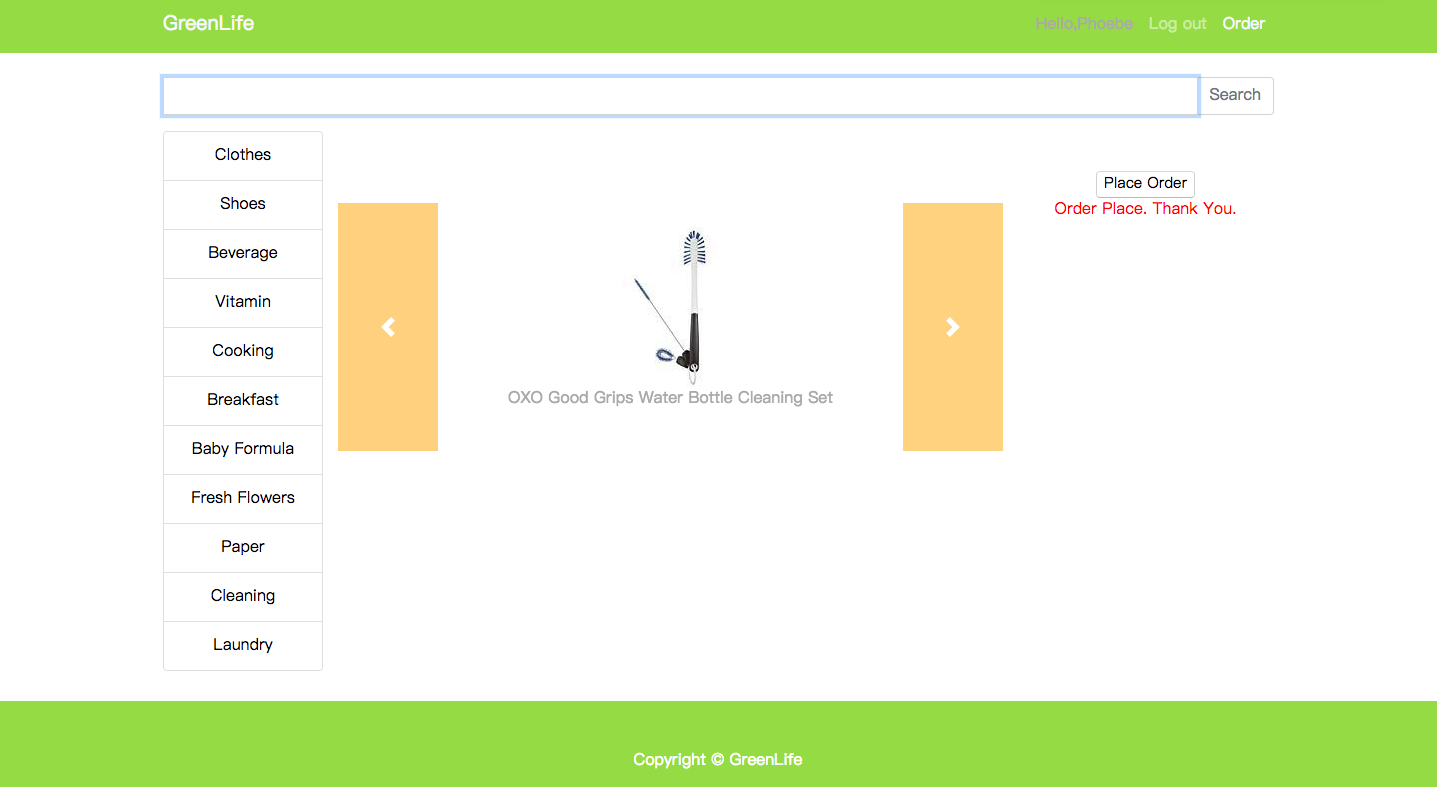


This page will provide detail of each order. Include order id, product id, product name, quality, price and time placed.

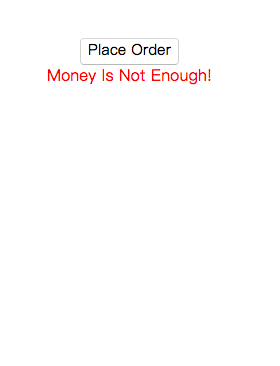
Testing:

**Place Order**

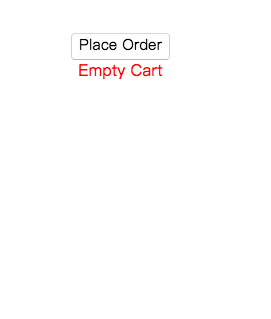
Have Finished shopping transaction:

****

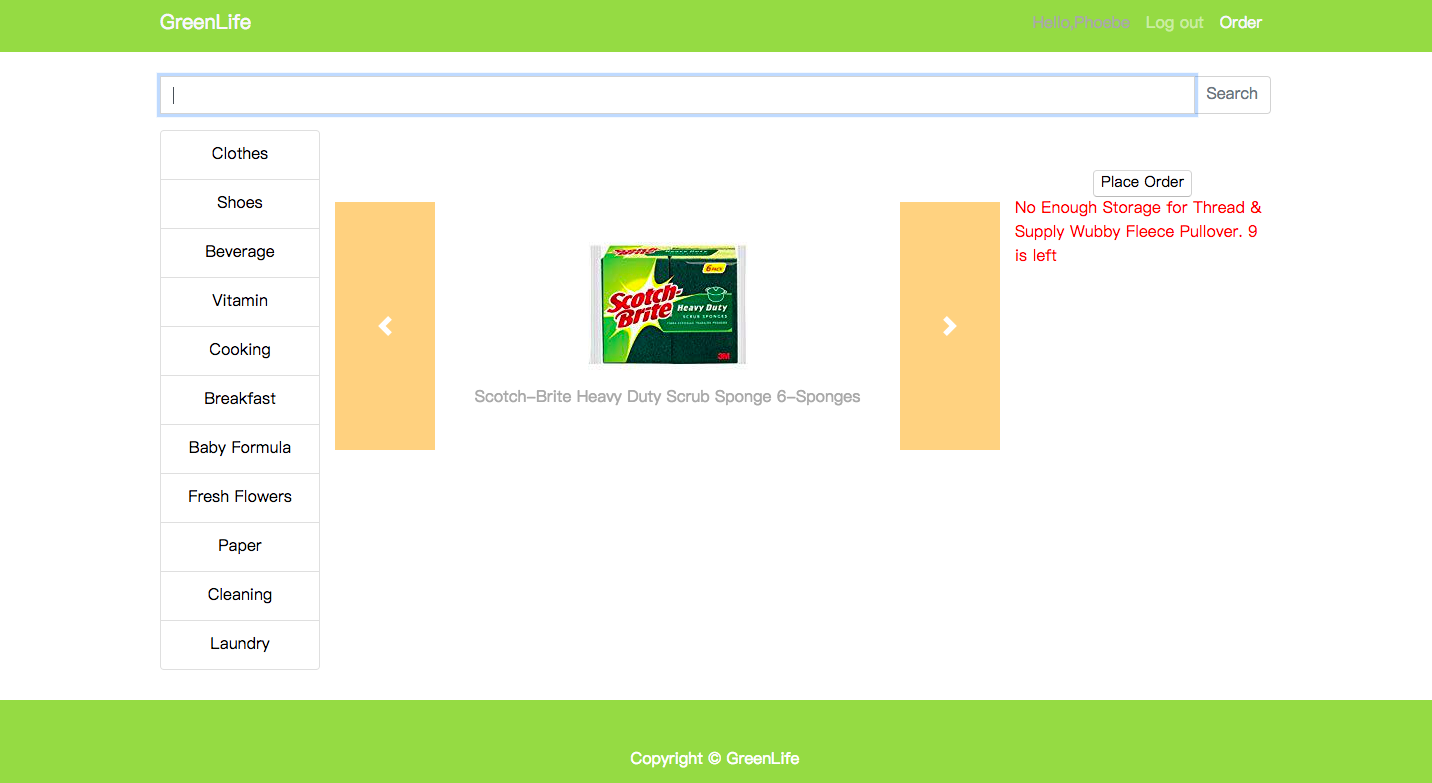
Do not have enough money to purchase items in cart:

****

Have not added any item to cart:

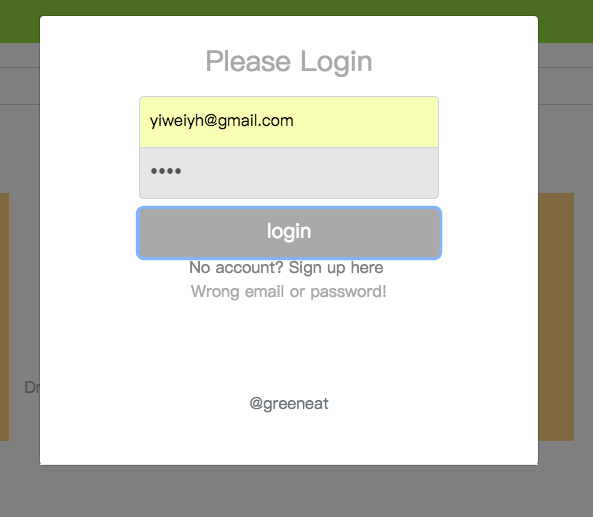
****

Do not have enough storage:

****

**Login**

Input wrong email or password:

****

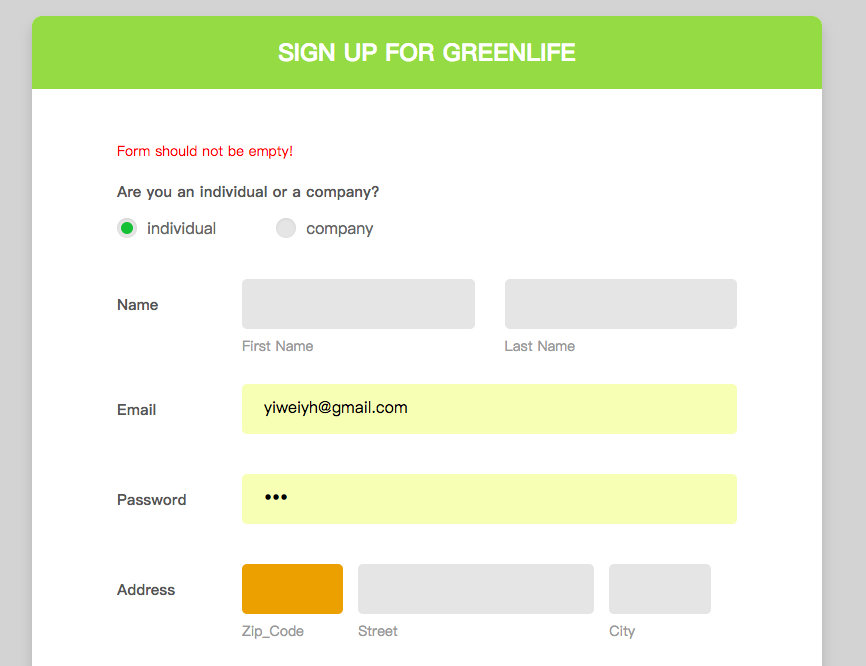
**Search**

Have not inputted anything or input one or more “space”:

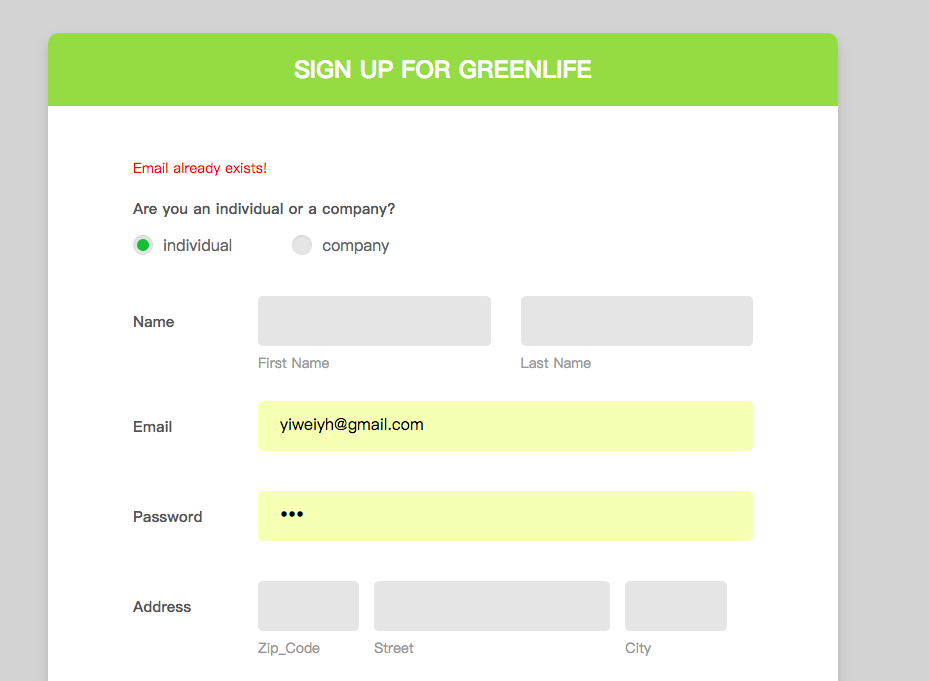
****

**Register**

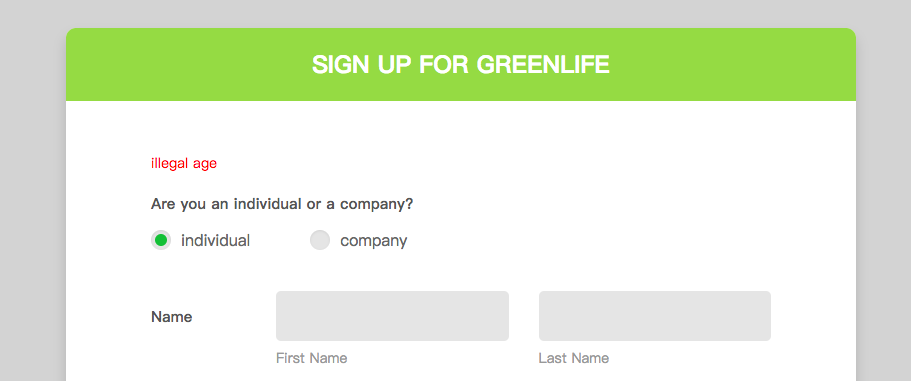
Have not filled in all the columns:



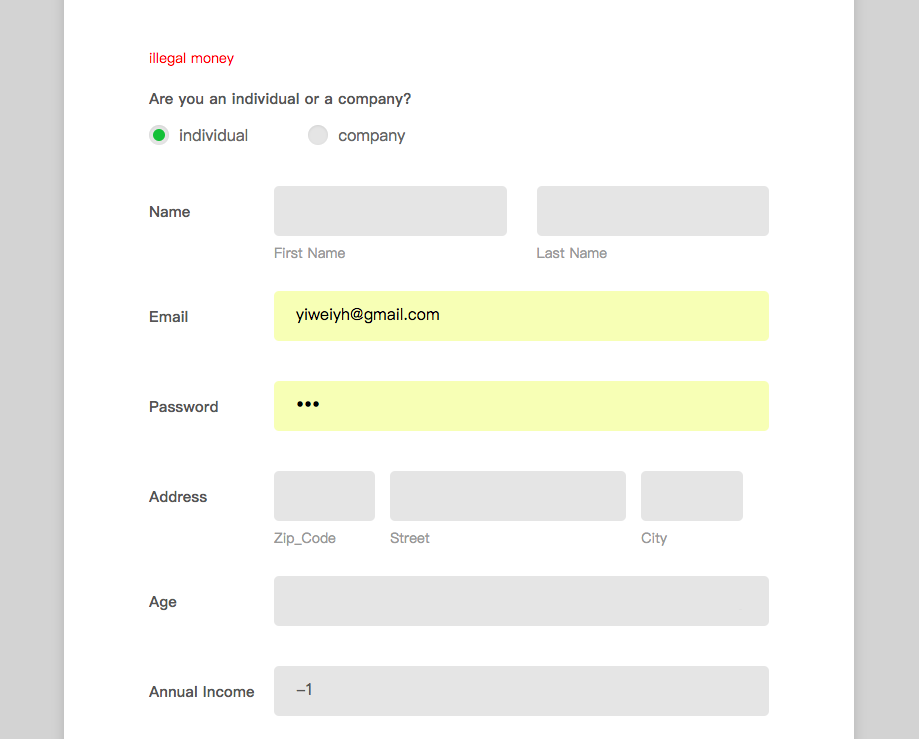
Have inputted duplicated email:



Have inputted an age number below 0:



Have inputted a number less below 0:



Limitation & Improvements

**Limitations:**

1. The system doesn’t have the payment function.
2. We cannot guarantee the stability of the system when a great number of users are online.
3. We cannot guarantee the security of user information.
4. No comment function

**Improvements:**

1. Create a comment function so that customer can add feedback here.
2. The implementation between client and server should be separate more thoroughly. Within the limited time, we did not process all the request and response by Ajax. But use Ajax can promote a better separation, and a better usage of different types of data.
3. In order to improve the robustness of the system, we should design cache from CDN cache to view layer cache to business logic cache to database cache to memory cache to CPU level 1 and level 2 cache.
4. In order to increase the concurrency of the system, we should use the CAS principle to build a lock free data structure.
5. To improve the security of the system. We should update security methods for registration, like adding security questions.