

INFSCI 2710 – Database Management – Fall 2018

Course Project-Mall Management System

TianHE ,TIH31@PITT.EDU

ChengRen ,CHR86@PITT.EDU

Swapnil Asawa ,SWA12@PITT.EDU

Overview

A shopping mall sells a great variety of goods. Managers of the shopping mall would like use a web application to manage all the data. They are also interested in some questions like, how many goods are sold; which goods need to be replenished; which kinds of goods are popular; which kinds customers spend more money in this mall, etc. To answer these questions, managers need to collect lots of data, such as, daily and seasonal sales for each kind of good; information of customers who buy certain types of goods; inventories for each types of goods; transactions details for every item sold, and so on. Since a shopping mall sells a great variety of goods, we need to create a database management system to record a mass of data and efficiently query from the data. Through the system, managers, who are also the administrators, would be able to perform add/update/view functions to transaction, customer,salesman,department,product, supplier and supply. Also, it allows managers to check the hottest products in certain period, replenishment requirement of products.

Assumptions

1. The Mall is consist of six departments, of which own different salesmen and products. One salesman can only belong to one department. One department can have many salesmen. So does the relationship between products and departments. We also have multiple customers, assuming all customer are members of this mall.
2. We have three suppliers and we keep historical supply record in supply table in which we create one entry per product_id provided by one supplier in one transaction. So that we can calculate aggregates by this table instead of creating another table.
3. We assume we cannot retrieve actual profit on one product but we can get the average. Because different sellers can supply same product at different price, but product with same id is sold in the shopping mall with one price only. Thus, we can get average profit on any kind of product but not profit on one product.
4. All people in our database should be any of the 2 genders.
5. We create a discount column for every product customer buy. So, we can adjust prices for certain product in two ways: The original price and discounts.
6. We connect both salesman and product to department to know details like which salesman is working best in any department.

7. We assume same product can have different product ids, ex.” honey” can be of multiple types brands etc.
8. Not considering about expiration date in this version
9. Every identical product is given same product id. I.e. 2 watches of xyz model of a company Sonata have same product_id. Because of that we cannot track one specific watch when it’s in the store.
10. We create dummy data to demonstrate the system.

E-R diagram

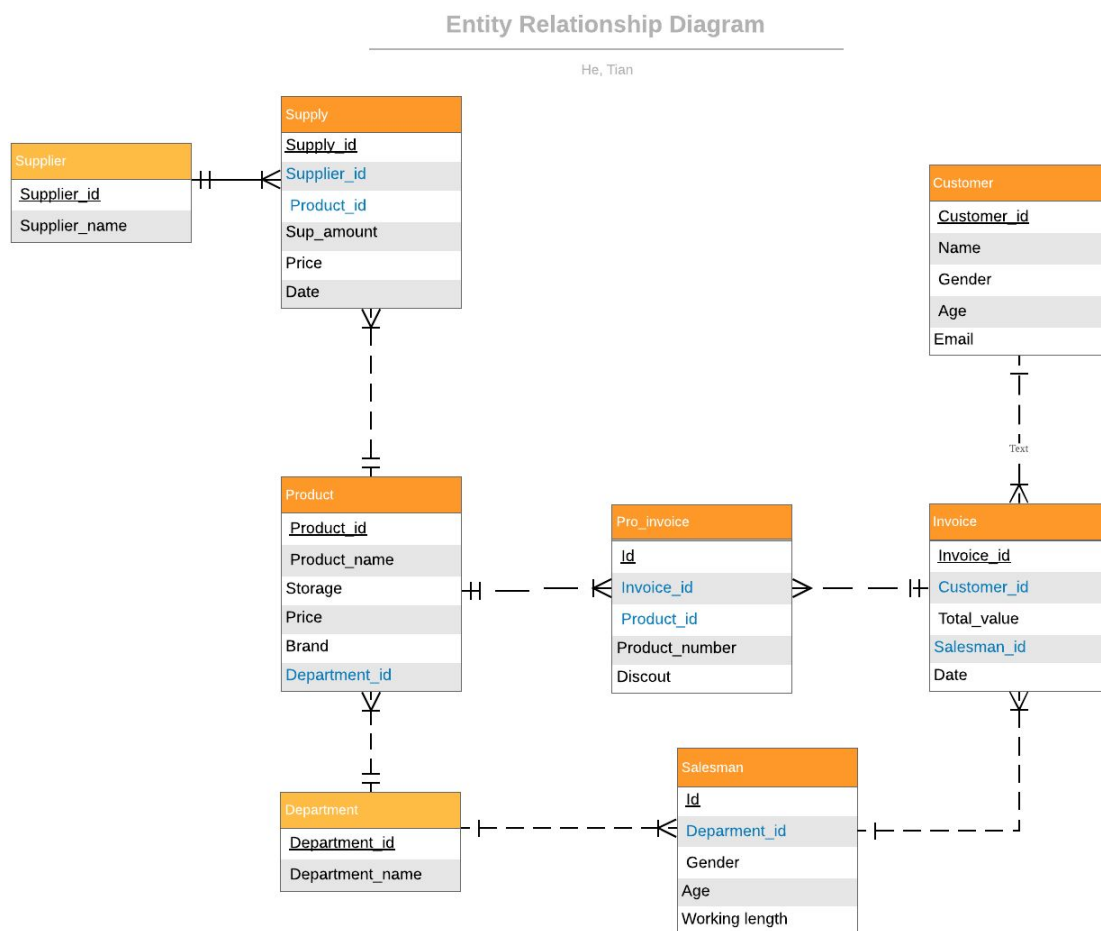


Table Customer, primary key:Customer_ID, which is foreign key of invoice table. A customer can have many invoices(orders). A invoice is only related one customer.

Table Supplier, primary key:Supplier_ID, which is foreign key of Supply table. One supplier would provide many supplies. One supply comes from one supplier

Table Department, primary key:Department_id, which is foreign key of product and salesman table. One department has different salesman. One salesman only belongs to one department. So does product.

Table salesman, primary key: id.which is foreign key of invoice table. One salesman will be responsible for several invoice. One invoice is completed by one salesman.

Table invoice, primary key:invoice_id,which is foreign key of pro_invoice table. pro_invoice is detail entity of invoice. One invoice is consist of several products.

Table Product, primary key:product_id, which is foreign key of supply and pro_invoice table. Relationship between supply and product is one to one. Product and pro_invoice is also one to one.

Table supply, primary key:supply_id.

Table pro_invoice, primary key: id.

DDL statement

```
CREATE TABLE IF NOT EXISTS `customer` (  
    `Customer_id` int(10) NOT NULL,  
    `name` varchar(50) NOT NULL,  
    `gender` varchar(10) NOT NULL,  
    `birthday` date DEFAULT NULL,  
    `email` varchar(250) DEFAULT NULL  
) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=latin1;  
  
CREATE TABLE IF NOT EXISTS `Department` (  
    `id` int(10) NOT NULL,  
    `name` varchar(50) NOT NULL  
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;  
  
CREATE TABLE IF NOT EXISTS `invoice` (  
    `Invoice_id` int(10) NOT NULL,  
    `Customer_id` int(10) NOT NULL,  
    `Total_value` float NOT NULL,  
    `salesman_id` int(10) NOT NULL,  
    `date` datetime DEFAULT NULL  
) ENGINE=InnoDB AUTO_INCREMENT=128 DEFAULT CHARSET=latin1;  
  
CREATE TABLE IF NOT EXISTS `Product` (  
    `Product_id` int(10) NOT NULL,  
    `Product_name` varchar(50) NOT NULL,  
    `Inventory` int(10) unsigned NOT NULL,  
    `Price` int(10) unsigned NOT NULL,  
    `Brand` varchar(50) NOT NULL,  
    `Department_id` int(11) NOT NULL  
) ENGINE=InnoDB AUTO_INCREMENT=37 DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `pro_invoice` (
  `id` int(10) NOT NULL,
  `invoice_id` int(10) NOT NULL,
  `product_id` int(10) NOT NULL,
  `Product_number` int(10) unsigned NOT NULL,
  `Discount` float NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=182 DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `Salesman` (
  `id` int(10) NOT NULL,
  `name` varchar(50) NOT NULL,
  `department_id` int(10) NOT NULL,
  `gender` varchar(10) NOT NULL,
  `birthday` date NOT NULL,
  `start_working_date` date NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `supplier` (
  `id` int(10) NOT NULL,
  `name` varchar(50) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `Supply` (
  `Supply_id` int(10) NOT NULL,
  `Supplier_id` int(10) NOT NULL,
  `Product_id` int(10) NOT NULL,
  `Sup_amount` int(10) NOT NULL,
  `Price` int(10) NOT NULL,
  `date` date DEFAULT NULL
) ENGINE=InnoDB AUTO_INCREMENT=66 DEFAULT CHARSET=latin1;
```

```
--
-- Indexes for table `customer`
--
ALTER TABLE `customer`
  ADD PRIMARY KEY (`Customer_id`);
```

```
--
-- Indexes for table `Department`
--
ALTER TABLE `Department`
  ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `name` (`name`);
```

```

--
-- Indexes for table `invoice`
--
ALTER TABLE `invoice`
  ADD PRIMARY KEY (`Invoice_id`),
  ADD KEY `cus_id_fk` (`Customer_id`),
  ADD KEY `sals_id_fk` (`salesman_id`);

--
-- Indexes for table `Product`
--
ALTER TABLE `Product`
  ADD PRIMARY KEY (`Product_id`),
  ADD KEY `Department_id_fk` (`Department_id`);

--
-- Indexes for table `pro_invoice`
--
ALTER TABLE `pro_invoice`
  ADD PRIMARY KEY (`id`),
  ADD KEY `invoice_id_fk1` (`invoice_id`),
  ADD KEY `Product_id_fk1` (`product_id`);

--
-- Indexes for table `Salesman`
--
ALTER TABLE `Salesman`
  ADD PRIMARY KEY (`id`),
  ADD KEY `de_id_fk` (`department_id`);

--
-- Indexes for table `supplier`
--
ALTER TABLE `supplier`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `Supply`
--
ALTER TABLE `Supply`
  ADD PRIMARY KEY (`Supply_id`),
  ADD KEY `Supplier_id_fk` (`Supplier_id`),
  ADD KEY `Product_id_fk` (`Product_id`);

--
-- AUTO_INCREMENT for table `customer`
--
ALTER TABLE `customer`
  MODIFY `Customer_id` int(10) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=16;
--

```

```

-- AUTO_INCREMENT for table `Department`
--
ALTER TABLE `Department`
  MODIFY `id` int(10) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=6;
--
-- AUTO_INCREMENT for table `invoice`
--
ALTER TABLE `invoice`
  MODIFY `Invoice_id` int(10) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=128;
--
-- AUTO_INCREMENT for table `Product`
--
ALTER TABLE `Product`
  MODIFY `Product_id` int(10) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=37;
--
-- AUTO_INCREMENT for table `pro_invoice`
--
ALTER TABLE `pro_invoice`
  MODIFY `id` int(10) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=182;
--
-- AUTO_INCREMENT for table `Salesman`
--
ALTER TABLE `Salesman`
  MODIFY `id` int(10) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=8;
--
-- AUTO_INCREMENT for table `supplier`
--
ALTER TABLE `supplier`
  MODIFY `id` int(10) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=4;
--
-- AUTO_INCREMENT for table `Supply`
--
ALTER TABLE `Supply`
  MODIFY `Supply_id` int(10) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=66;
--
-- Constraints for table `invoice`
--
ALTER TABLE `invoice`
  ADD CONSTRAINT `cus_id_fk` FOREIGN KEY (`Customer_id`) REFERENCES
`Customer` (`Customer_id`),
  ADD CONSTRAINT `sals_id_fk` FOREIGN KEY (`salesman_id`) REFERENCES
`Salesman` (`id`);
--
-- Constraints for table `Product`
--
ALTER TABLE `Product`
  ADD CONSTRAINT `Department_id_fk` FOREIGN KEY (`Department_id`)
REFERENCES `Department` (`id`);

```

```

--
-- Constraints for table `pro_invoice`
--
ALTER TABLE `pro_invoice`
  ADD CONSTRAINT `Product_id_fk1` FOREIGN KEY (`product_id`) REFERENCES
`Product` (`Product_id`),
  ADD CONSTRAINT `invoice_id_fk` FOREIGN KEY (`invoice_id`) REFERENCES
`Invoice` (`Invoice_id`),
  ADD CONSTRAINT `invoice_id_fk1` FOREIGN KEY (`invoice_id`) REFERENCES
`Invoice` (`Invoice_id`);

--
-- Constraints for table `Salesman`
--
ALTER TABLE `Salesman`
  ADD CONSTRAINT `de_id_fk` FOREIGN KEY (`department_id`) REFERENCES
`Department` (`id`);

--
-- Constraints for table `Supply`
--
ALTER TABLE `Supply`
  ADD CONSTRAINT `Product_id_fk` FOREIGN KEY (`Product_id`) REFERENCES
`Product` (`Product_id`),
  ADD CONSTRAINT `Supplier_id_fk` FOREIGN KEY (`Supplier_id`) REFERENCES
`supplier` (`id`);

```

Front-end & Back-end and System Overview

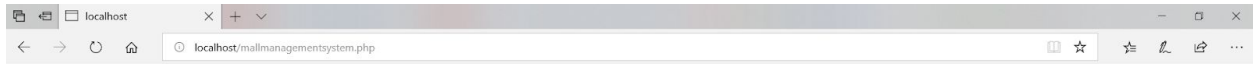
Front-end tools: PHP, Javascript

Front-End Back-End communication method: MySQLi, AJAX

Startpage: MallManagementSystem.php

The MySQLi Extension (MySQL Improved) is a relational database driver used in the PHP scripting language to provide an interface with MySQL databases. Through MySQLi connection, PHP is able to get data from MySQL database and the data customers input can be import to database.

Front end design starting from frameset to show eight sections at left of the portal and the content of sections will be shown on the right. The main page shows four aggregate functions which are used frequently. The other seven sections are detail management of each table. On each page, we can add new data to database, retrieve required information from database, edit existed data in database through front end.



Mall Management System

[Main Page](#)

[Transaction Management](#)

[Department Management](#)

[Sales Management](#)

[Customer Management](#)

[Product Management](#)

[Supplier Management](#)

[Supply Management](#)

Hottest search

Top 5 hottest Product from to

[Last Six Month Total Sales](#)

[Aggregate Profit](#)

[Restock Checking](#)

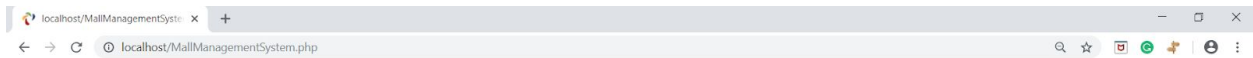


Aggregate functions

We have four aggregate functions which allow analyze the data.

1.Aggregate Profit:

Enter dates to check overall profit in the range:



Mall Management System

[Main Page](#)

[Transaction Management](#)

[Department Management](#)

[Sales Management](#)

[Customer Management](#)

[Product Management](#)

[Supplier Management](#)

[Supply Management](#)

Check overall profit

From Date:

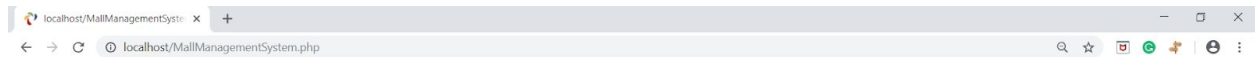
To Date:

See Profit from a product

From Date:

To Date:

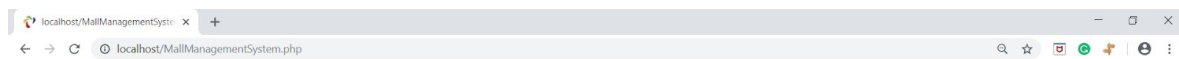




Main Page
Transaction Management
Department Management
Sales Management
Customer Management
Product Management
Supplier Management
Supply Management

The total profit from: 2018-01-01 to 2018-07-19 is 3119.30

On checking product profit, with a particular range. It will show profit from each product which was sold in that date range, in decrease order of profit. This can answer questions like max profit producing objects.



Main Page
Transaction Management
Department Management
Sales Management
Customer Management
Product Management
Supplier Management
Supply Management

Products Profit

Product ID	Product Name	Average Supply Price	Average Selling Price	Quantity	Total Profit
14	macbookpro	1400.0000	1890.00	6	2940.00
1	Honey	12.0000	16.58	9	41.26
6	Milk	11.0000	14.97	7	27.80
4	Cheetos	11.0000	15.20	6	25.20
5	Tomato	12.0000	15.92	6	23.54
8	butter	7.0000	9.39	7	16.70
12	ketchup	8.0000	11.26	5	16.28
10	tegreen	7.0000	9.72	5	13.60
2	Potatos	9.0000	11.96	2	5.92
3	Honey	11.0000	13.80	2	5.60
7	bread	11.0000	14.40	1	3.40

Profit per product in the given date range, order by max to min profit giving product.
Take the average cost price from purchases from supplier.
Take the average selling price from sells to customers in the specified date range.

Subtract avg_selling_price from avg_cost_price to get avg profit per product sold.
Multiply it by quantity of that product sold in the date range to get the total profit of the product in the range.
Group by these result for all the products
Order by profit in descending order

Stored procedure :

```
CREATE PROCEDURE check_productwise_profit(IN start_date date, IN end_date date)
BEGIN
    Select cost.Product_id, Product_name, costp as
    average_cost_price, avg_selling_price, quant as quantity_sold_between_given_date,
    (avg_selling_price- costp)*quant as product_total_profit from (Select avg(price) as
    costp, Product_id as Product_id from Supply group by Product_id) as cost, (Select
    pi.product_id as pid, p.Price, p.Product_name as Product_name, count(pi.discount) as
    quant, avg(pi.discount) as disc, p.Price*avg(pi.discount) as avg_selling_price from
    Product p, pro_invoice pi, invoice as i where pi.product_id=p.Product_id and
    i.Invoice_id =pi.invoice_id and i.date >start_date and i.date< end_date Group by
    p.product_id) as sell Where cost.Product_id=sell.pid order by product_total_profit desc;
END
```

To get the total profit in a date range, the result of the above sql query is added like this:

Stored procedure :

```
CREATE PROCEDURE check_total_profit(IN start_date date, IN end_date date)
BEGIN
    Select sum((avg_selling_price- costp)*quant) as total_profit from (Select
    avg(price) as costp, Product_id as Product_id from Supply group by Product_id) as
    cost, (Select pi.product_id as pid, p.Price, count(pi.discount) as quant, avg(pi.discount)
    as disc, p.Price*avg(pi.discount) as avg_selling_price from Product p, pro_invoice pi,
    invoice as i where pi.product_id=p.Product_id and i.Invoice_id =pi.invoice_id and i.date
    >start_date and i.date< end_date Group by p.product_id) as sell Where
    cost.Product_id=sell.pid;
END
```

When you don't enter start or end date, it will show you results from the 1st recorded date to last recorded date. On checking overall profit without entering date:

Mall Management System

[Main Page](#)

The total profit from: 1900-01-01 to 2200-01-01 is 5543.75

[Transaction Management](#)

[Department Management](#)

[Sales Management](#)

[Customer Management](#)

[Product Management](#)

[Supplier Management](#)

[Supply Management](#)

On checking product wise profit without entering date:

The result is sorted according to the most profit making products to least.

Mall Management System

[Main Page](#)

[Transaction Management](#)

[Department Management](#)

[Sales Management](#)

[Customer Management](#)

[Product Management](#)

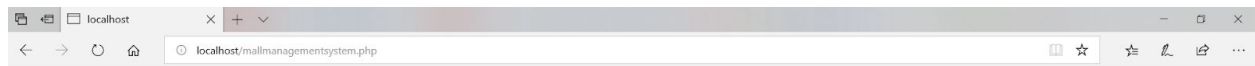
[Supplier Management](#)

[Supply Management](#)

Products Profit

Product ID	Product Name	Average Supply Price	Average Selling Price	Quantity	Total Profit
14	macbookpro	1400.0000	1914.00	10	5140.00
1	Honey	12.0000	16.42	17	75.14
5	Tomato	12.0000	16.11	13	53.44
4	Cheetos	11.0000	15.10	12	49.60
8	butter	7.0000	9.52	16	45.40
6	Milk	11.0000	15.11	11	45.24
12	kelpod	8.0000	11.28	12	39.36
3	Honey	11.0000	14.02	10	30.25
7	bread	11.0000	14.31	8	26.45
2	Potatos	9.0000	12.24	7	22.67
10	tiegreen	7.0000	9.70	6	16.20

2) When you enter start date less than end date in overall profit, You get the error that it is not allowed and reenter.



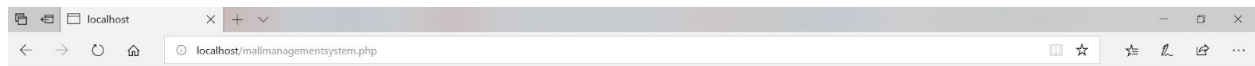
end date cannot be smaller than start date. Enter correctly

[Main Page](#)
[Transaction Management](#)
[Department Management](#)
[Sales Management](#)
[Customer Management](#)
[Product Management](#)
[Supplier Management](#)
[Supply Management](#)

http://localhost/check_total_profit.php

2.Hottest Products

Enter the begin date and end date will get top 5 hottest products in this period.



Mall Management System

[Main Page](#)
[Transaction Management](#)
[Department Management](#)
[Sales Management](#)
[Customer Management](#)
[Product Management](#)
[Supplier Management](#)
[Supply Management](#)

Best Seller from 2018-08-03 to 2018-12-03

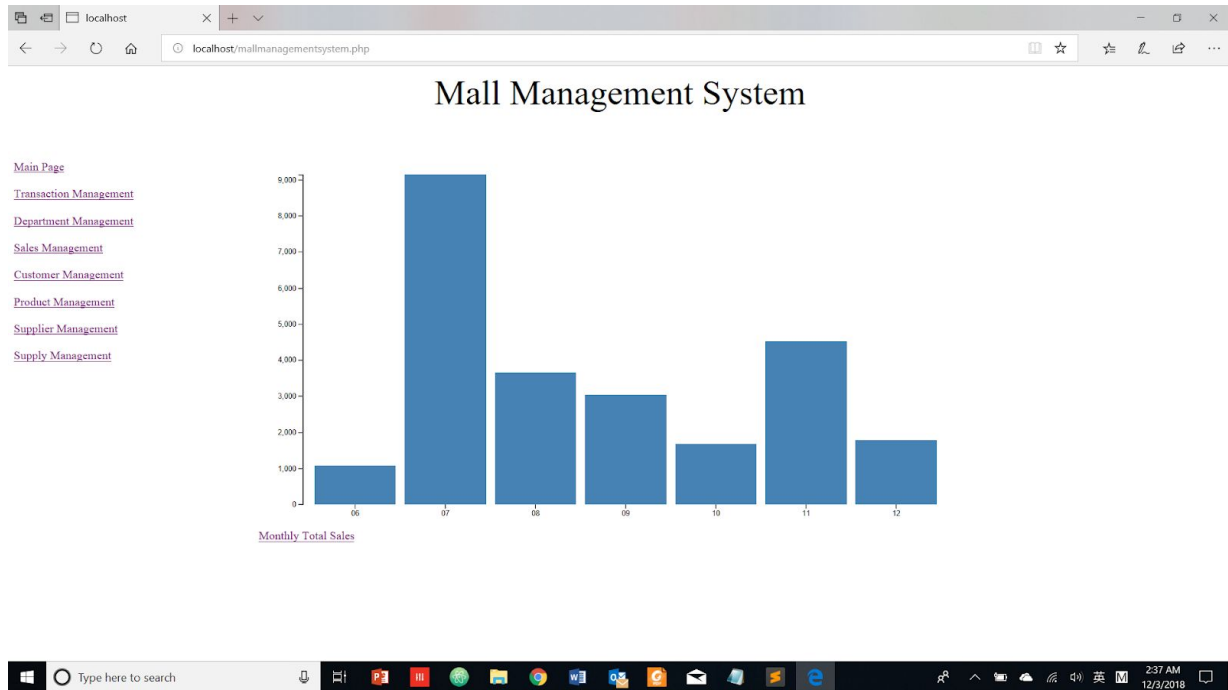
Ranking	Product	Quantity of Sales
1	butter	10
2	Honey	7
3	bread	7
4	geo10th	6
5	dressredblue	6

[Back](#)

Type here to search

3.Last six month sales amount.

A bar graph will be shown to demonstrate the sales amount in last six month.



4. Restock checking

This function is to check the low inventory level products. We calculate the safety stock as the monthly average demand of last 12 month. Also the replenishment is equal to the safety stock.

Mall Management System

Restock

Product_id	Inventory level	Safety Stock	Replenishment
1	4	5	low stock level
15	4	5	low stock level

Transaction Management:

There are two parts in Transaction Management

1.Create New Transaction

2.Review Transactions

The screenshot shows a web browser window with the address bar displaying 'localhost/MallManagementSystem.php'. The page title is 'Mall Management System'. On the left, there is a sidebar menu with links: Main Page, Transaction Management, Department Management, Sales Management, Customer Management, Product Management, Supplier Management, and Supply Management. The main content area is divided into two sections. The top section, 'Create New Transaction', contains two input fields: 'CustomerID' and 'SalesmanID', followed by a 'Create Order' button. The bottom section, 'Review Transactions', contains an 'Invoice' input field and a 'Search Order' button. The Windows taskbar at the bottom shows the search bar, taskbar icons, and system tray with the date '12/2/2018' and time '6:31 PM'.

1.Create New Transaction. Customer and Salesman ID are required to input to create a new order, otherwise error message will be pop up.

This screenshot shows the same web application as the previous one, but with an error message dialog box displayed in the center. The dialog box has a title 'localhost says' and a message 'Please input Customer ID and Salesman ID.' with an 'OK' button. The background content of the page is partially obscured by the dialog box. The Windows taskbar at the bottom shows the search bar, taskbar icons, and system tray with the date '12/2/2018' and time '6:34 PM'.

Create a new order for customer id 1,salesman id 1

localhost/MallManagementSystem x +

localhost/MallManagementSystem.php

Mall Management System

Main Page
Transaction Management
Department Management
Sales Management
Customer Management
Product Management
Supplier Management

Customer Info

Customer id	name	Gender	birthday	Email
1	Graggi	male	1985-09-04	datafa@id.com

Salesman Info

Salesman ID	name	Department ID	Gender	Birthday	Start working from
1	Sam	1	M	1985-09-04	2000-02-01

Input Detail for Order No.147

Product ID	Quantity	Discount

Save Add

Transaction History

Invoice_id	Product_id	Quantity	Discount
144	5	1	1.00
144	1	5	1.00
144	1	5	1.00

Type here to search

6:34 PM 12/2/2018

Order detail are permitted to input multiple rows, all fields are required to input. If the Quantity is lower than inventory, error message will be popped up.

localhost/MallManagementSystem x +

localhost/MallManagementSystem.php

localhost says
Inventory lower than requestInventory lower than request

OK

Customer Info

Customer id	name	Gender	birthday	Email
1	Graggi	male	1985-09-04	datafa@id.com

Salesman Info

Salesman ID	name	Department ID	Gender	Birthday	Start working from
1	Sam	1	M	1985-09-04	2000-02-01

Input Detail for Order No.147

Product ID	Quantity	Discount

Delete Add

Save

Transaction History

Invoice_id	Product_id	Quantity	Discount
144	5	1	1.00
144	1	5	1.00

Type here to search

6:37 PM 12/2/2018

After input order detail for this order, feedback would be the total value of this order. Then the transaction history will be automatically updated.

2.Review Transaction.

Input Order number. If the order number are not exist, error message will be popped up, otherwise invoice detail will be shown below.

The screenshot shows a web browser window with the address bar displaying 'localhost/MallManagementSystem.php'. The page title is 'Mall Management System'. On the left, there is a sidebar menu with links: Main Page, Transaction Management, Department Management, Sales Management, Customer Management, Product Management, Supplier Management, and Supply Management. The main content area has three sections: 'Create New Transaction' with input fields for 'CustomerID' and 'SalesmanID' and a 'Create Order' button; 'Review Transactions' with an 'Invoice' input field and a 'Search Order' button; and 'Invoice Detail' which displays a table with the following data:

Invoice ID	Customer	Total value	Salesman	Date	
22	9	102	4	2018-12-28 00:00:00	View

At the bottom of the browser window, the Windows taskbar is visible, showing the search bar and various application icons. The system clock indicates 6:49 PM on 12/2/2018.

To view the detail, we can also update new item to this order.

The screenshot shows the same web browser window, but the 'Update Order' section is active. It features a table titled 'Order Detail' with the following data:

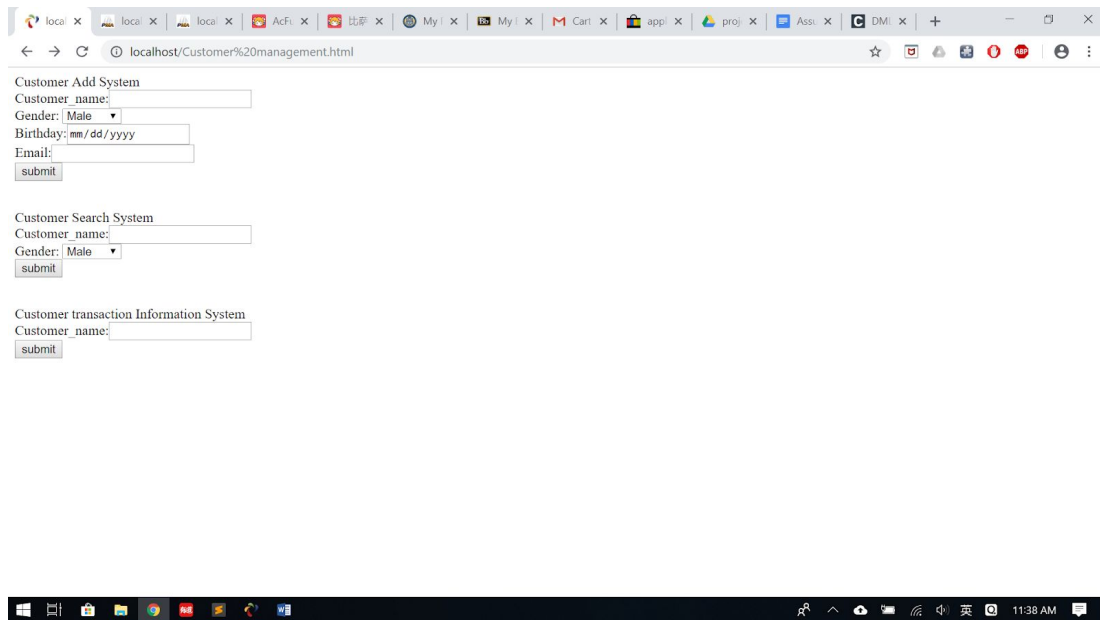
Invoice_id	Product_id	Quantity	Discount
22	7	5	1.00
22	12	1	1.00
22	3	1	1.00
22	17	1	0.99

Below the table, there is an 'Update Order' section with input fields for 'Product ID', 'Quantity', and 'Discount', and an 'Add Transaction' button. At the bottom, a message states: 'Total value of order NO 22 is 104.97'. The Windows taskbar at the bottom shows the system clock as 6:51 PM on 12/2/2018.

Customer management:

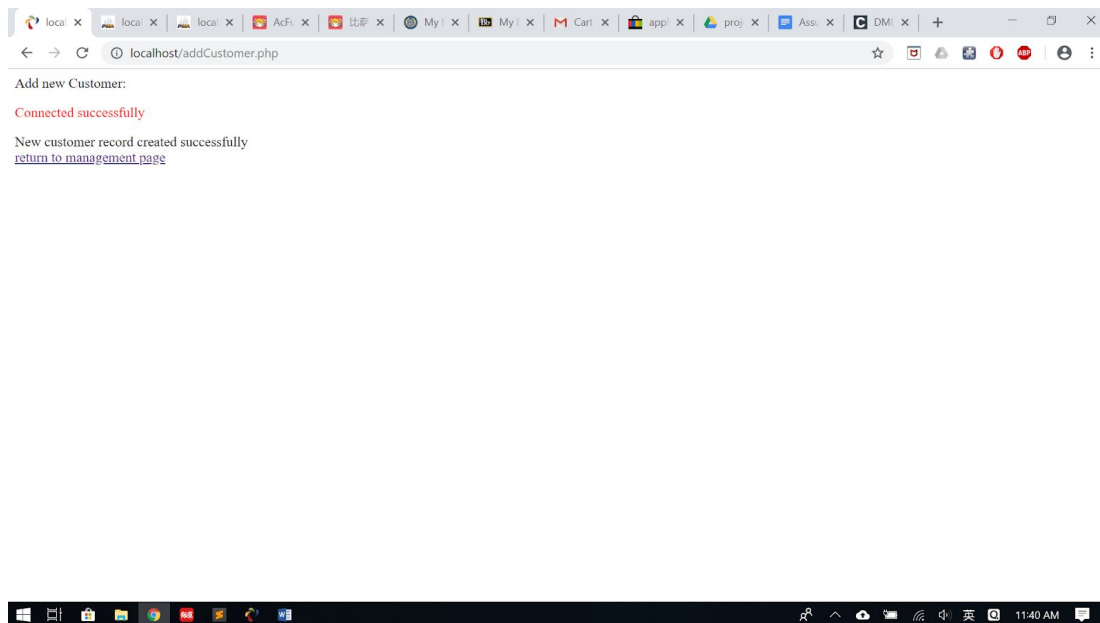
We have three functions in Customer management:

1. Customer Add System
2. Customer Search System
3. Customer transaction Information System

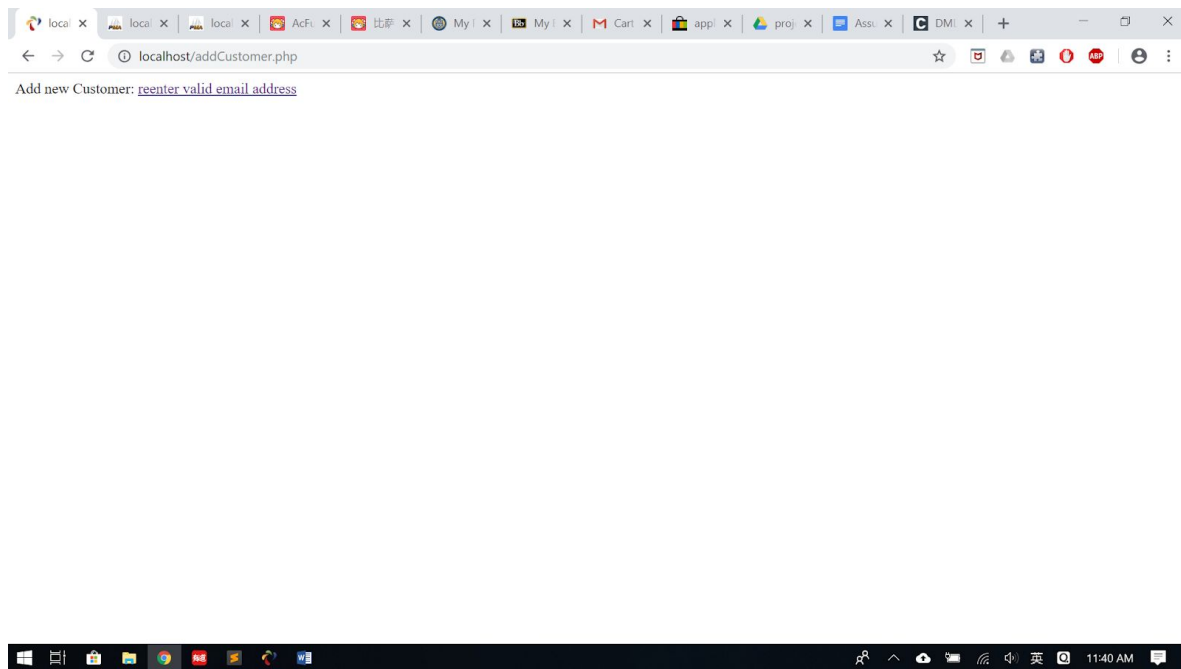


1. Customer Add System

If customer is added successfully:

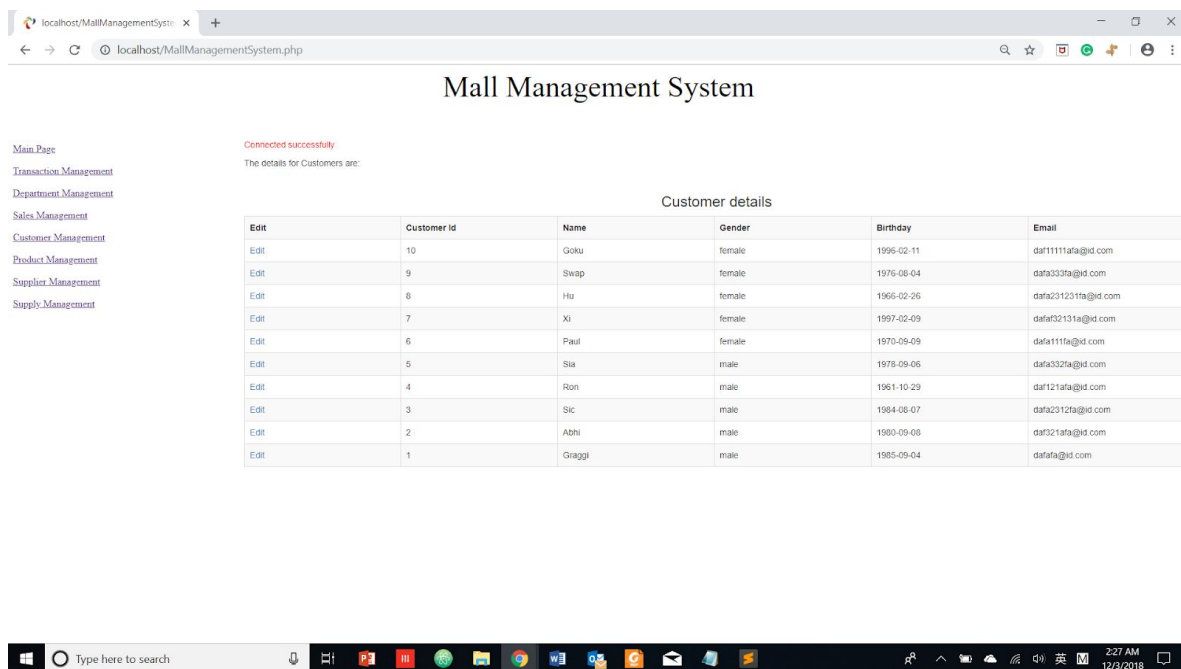


If the email address is invalid (the email address can be null):



2. Customer Search System

If enter a name or letters, the system will return all customers whose names contain these letters. If we do not enter customer name, it will defaultly return 10 newly added customers. If there is no customer found, you will get a message that no customer found.



We can click 'Edit' to change customer's information

localhost x local x local x AcFi x 比菲 x My i x My i x Cart x appi x proj x Assu x DMi x + - X

localhost/editCustomer.php?ID=1

update Customer's information:

Connected successfully

Name: Graggi

Gender: male

Birthday: 09/04/1985

Email: dafafa@id.com

submit

Windows taskbar: 11:44 AM

3. Customer transaction Information System

If do not enter a customer name, it will give you warning that to re enter valid customer name .If there's no customer found you will get same warning.

localhost x local x local x AcFi x 比菲 x My i x My i x Cart x appi x proj x Assu x DMi x + - X

localhost/transactionCustomer.php

Connected successfully

Please enter valid customer name

Windows taskbar: 11:55 AM

If enter a name or letters, the system will return all customers whose names contain these letters and the amount of money they spent in our shopping mall

The screenshot shows a web browser window with the URL `localhost/mailmanagementsystem.php`. The page title is "Mall Management System". On the left, there is a sidebar with navigation links: [Main Page](#), [Transaction Management](#), [Department Management](#), [Sales Management](#), [Customer Management](#), [Product Management](#), [Supplier Management](#), and [Supply Management](#). The main content area displays a message "Connected successfully" and "The details for customer are:". Below this, a table titled "Consumption details" shows the following data:

details	Spending amount	Customer_id	Name
details	2033.460000038147	8	Hu
details	5730.810098648071	2	Abhi

The browser's taskbar at the bottom shows the Windows Start button, a search bar, and various application icons. The system clock indicates 2:31 AM on 12/3/2018.

Click details, we can see the details for transactions of one certain customer

The screenshot shows the same web browser window as the previous one, but with the "details" link clicked. The page title remains "Mall Management System". The sidebar navigation links are the same. The main content area now displays "The transaction details for customer are:". Below this, a table titled "Consumption details" shows the following data:

product id	product name	brand	price	number	discount	total money	date
3	Honey	BigB	15	3	0.94	42.29999989271164	2018-07-23 00:00:00
15	history9th	ncert	400	3	0.96	1151.9999742507935	2018-07-23 00:00:00
8	butter	BigC	10	2	0.95	18.99999976158142	2018-07-23 00:00:00
5	Tomato	BigC	17	3	0.98	49.9800009727478	2018-07-23 00:00:00
5	Tomato	BigC	17	2	0.98	33.320000648498535	2018-07-23 00:00:00
12	kelpod	medcure	12	3	0.9	32.399999141693115	2018-07-23 00:00:00
13	macbook	mac	1000	3	0.91	2730.000078678131	2018-07-23 00:00:00
3	Honey	BigB	15	2	0.9	26.999999284744263	2018-07-23 00:00:00
12	kelpod	medcure	12	3	0.93	33.480000257492065	2018-07-23 00:00:00
1	Honey	BigB	17	3	0.97	49.470001459121704	2018-07-23 00:00:00
14	macbookpro	mac	2000	3	0.98	5880.000114440918	2018-07-23 00:00:00
8	butter	BigC	10	2	0.91	18.200000524520874	2018-04-29 00:00:00
8	butter	BigC	10	2	0.95	18.99999976158142	2018-04-29 00:00:00
8	butter	BigC	10	2	0.93	18.600000143051147	2018-04-29 00:00:00

The browser's taskbar at the bottom shows the Windows Start button, a search bar, and various application icons. The system clock indicates 2:31 AM on 12/3/2018.

The functions of salesman/department/product/supplier/supply management are similar and simpler to customer management. So we do not show them in this report.

Limitation

1. We create dummy data to demonstrate how the system works, so any conclusion from the system are not meaningful.
2. From the point view of supply chain management, the safety stock should be an additional quantity of an item held in inventory in order to reduce the risk that the item will be out of stock. Safety stock acts as a buffer in case the sales of an item are greater than planned and/or the supplier is unable to deliver additional units at the expected time. Due to the limited information we created, we only consider the safety stock as the monthly average demand of last 12 month.
3. Login system hasn't been established.