Project Final Report

-FAMLOG-

CS157A - Section 01

Team 39

Benjamin Wen Shen Lee

Grace To

Jeffrey Nguyen

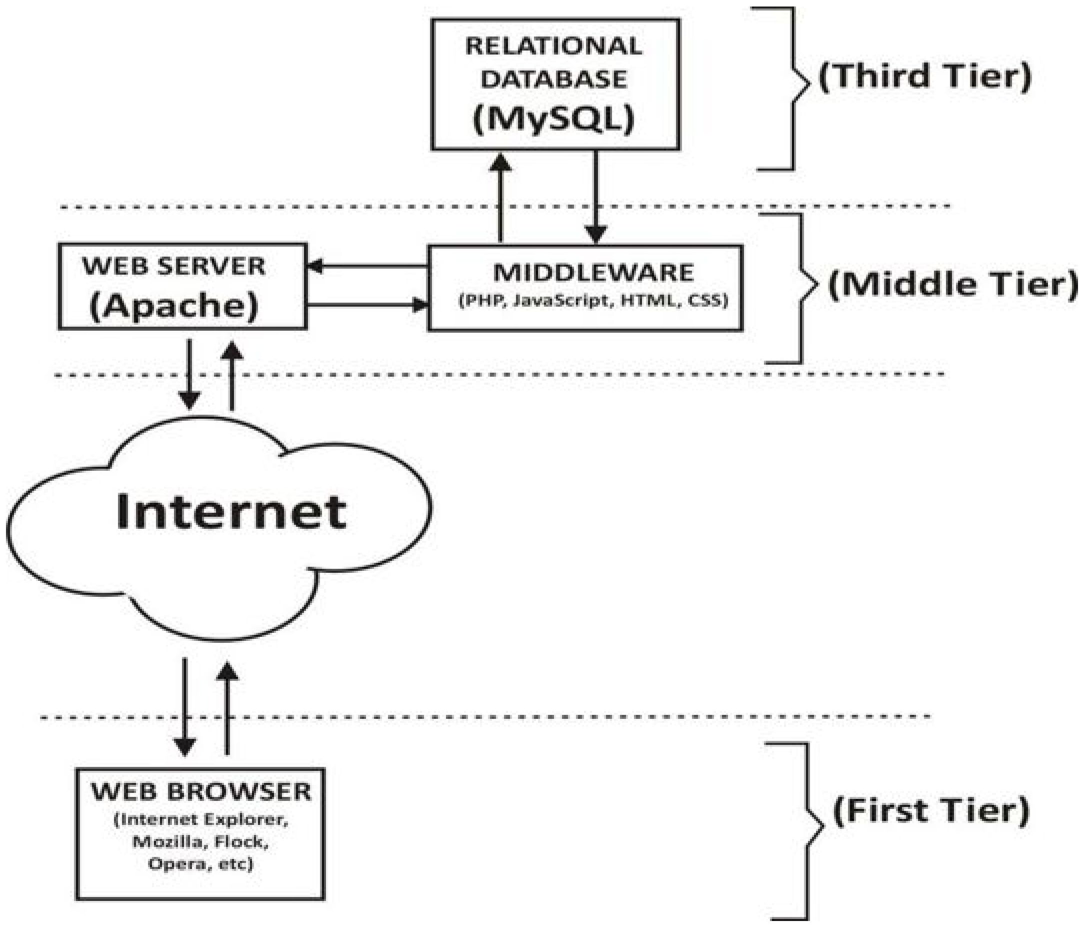
<u>Project Requirement</u>

Project Description

The goal of this project is to build a web application that will allow households to maintain a centralized shopping list based upon the aggregated, separate lists created by the individuals. The motivation is the facilitation of household shopping by giving all users under a household a view of the entire prioritized household shopping list while sustaining the personal lists. It will aid both the household as a whole and the individual user in the overall process of shopping for necessities and for groceries when others might be busy. The stakeholders of this product are the households with a number of users greater than or equalled to two that utilize this application. All those involved in the development of the project, such as the engineers and project managers, are also stakeholders. For this particular project, the application domain will be strictly maintained to the scope of a household. Many people have extraordinarily busy schedules, where even finding time to shop is difficult. The question we posed was why not have a list where a family or group of co-inhabitants have the ability to shop for others when they are out. The benefits to the users will be the ability for the individual user to view and to purchase items on the master list for others when they may already be shopping. The traditional personal shopping list will also be maintained should the user just necessitate their own shopping list.

System Environment

Structured diagram


RELATIONAL DATABASE (MySQL)
(Third Tier)
WEB SERVER (Apache)
MIDDLEWARE (PHP, JavaScript, HTML, CSS)
(Middle Tier)
Internet
WEB BROWSER (Internet Explorer, Mozilla, Flock, Opera, etc)
(First Tier)

Hardware and Software used

- Windows 10
- Apache
- Sublime Text
- PHPStorm
- Eclipse

RDBMS

- MySQL Community Server 8.0.17

Application Languages

- HTML, CSS, JavaScript, XML, PHP, SQL

Functional Requirements

- Description
    - The application target each household as a unit. Each household can access the system by logging in with just one account. Each household may have multiple users as individuals. Each user can choose to access between personal list and master list. Each user has their own wanted shopping list of items. In the personal list, user can add his/her wanted items into the shopping list. User can also edit his/her items in their own shopping list (remove items/edit notes/edit name). In the master list, user can see the total shopping items of everyone in the house in descending priority order. In the master list, the person who purchased the items can select and enter the price they bought the items for. The application will calculate the total price of the items bought (the selected items will be automatically removed from the list). User can log out of the system

- Functions
  - Creating an account
    - Each household only needs to sign up for one account
    - Completion of the sign up form is necessary to successfully create an account
  - Creating users
    - In each household account, user can add their name if its non-existent within that account
  - Create Personal Shopping List
    - Each user has their own wanted items shopping list automatically created together with their user creation
  - Display Personal Shopping List
    - User can view all the items in their personal shopping list
    - Sorting will also be available according to item priority
  - Edit Personal Shopping List
    - User can edit item name, notes, quantity, priority, or even remove the item
  - Display Master List view
    - Master list combines personal lists of all users in the household
    - Master list displays priority of items, items' details and quantity as well as whom it belongs to
  - Order Master List items by priority
    - Master list view of items are ordered according to priority
  - Select items purchased and enter items' prices
    - In the Master List, user selects all items that are bought and enters the prices
    - The items selected as purchased will be deleted from database
  - Calculate the total price + tax
    - The total price including the tax implied will automatically be calculated once the user has finished his/her selection

Non-functional Issues

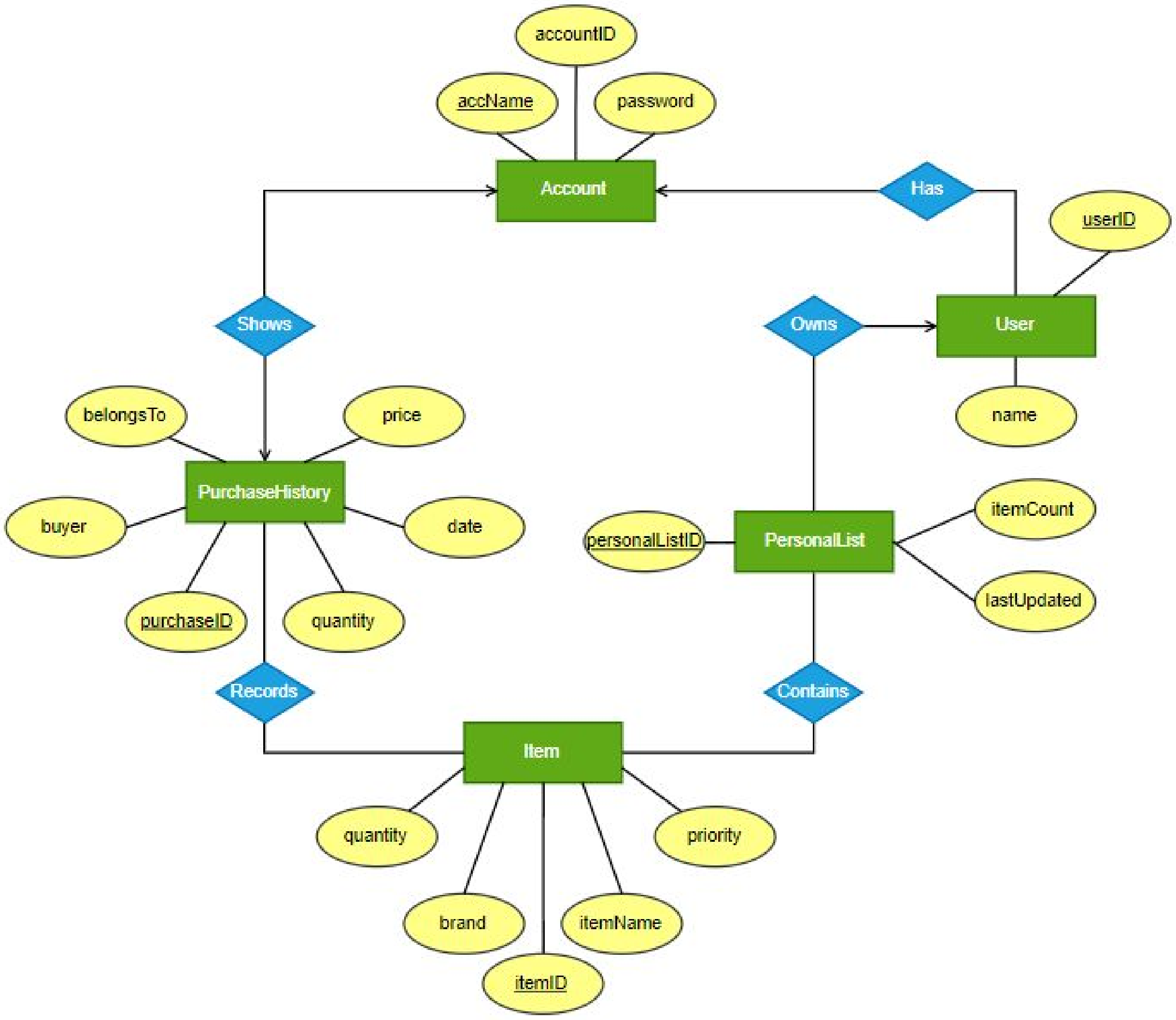
- Graphical User Interface
    - We plan to use Adobe XD to design the website interface and Adobe Photoshop to create necessary graphics. The interface should be minimal and easy to navigate, with clear selections to perform tasks. Firstly, there should be a login page, then a user selection. The user could then choose between Master List and Personal Shopping List, and the rest of the functions should be accessible through those screens.

- Security
    - Each household will have an account of its own. For more enhanced security, we might implement a passcode system for the users within that account, to make sure there is no misuse among users.

- Access Control
    - Our website should be available 24/7 on any browser, as long as users have access to the internet, and have also made an account for their household. Users of different accounts should also be able to access our website simultaneously without any interruptions while the other users are online. Users are also only allowed to alter information within their own household/user account.

<u>Project Design</u>

E/R Diagram



Database Schema

Account(accountID, accName, password)

Has(accName, userID)

User(userID, name)

Owns(userID, personalListID)

PersonalList(personalListID, itemCount, lastUpdated)

Contains(personalListID, itemID)

Item(itemID, itemName, brand, quantity, priority)

Records(itemID, purchaseID)

PurchaseHistory(purchaseID, belongsTo, buyer, quantity, date, price)

Shows(accName, purchaseID)

Descriptions

Through our application, any individual can create an *Account* with a unique *accountName* and a *password* for the respective household. A *User* is then created within the *Account*, representing the different individuals within that household.

Every *User* will have a unique *userID* and *name*. Each *User* also has exactly one *PersonalList* which is identified by a unique *personalListID*.

The entity *PersonalList* also contains the attributes which include the *quantity* requested, the *priority* of each item, and the attribute *notes*. This is representative of the individual shopping list, held by each *User* within a household.

PersonalList also has a relationship with the entity *Item*. *PersonalList* contains the *Item* that was chosen by the *User* to add to their list. *Item* has the attributes of *brand, itemName,* and the unique attribute *itemID*. *Item* shares a relationship with *PurchaseHistory. PurchaseHistory* records the *Item*, adding it to the rest of the purchase history.

By default, the *Account* has exactly one *PurchaseHistory*, and each *PurchaseHistory* belongs to exactly one *Account*. This entity has the unique attribute *purchaseID*. Other attributes of *PurchaseHistory* include the *buyer* of the item, the attribute *itemName*, the *date* of purchase, the *price* of each item, the *quantity* of the items bought, and the attribute *belongsTo*, which represents to whom's list the *Item* belongs to.

Tables

Account

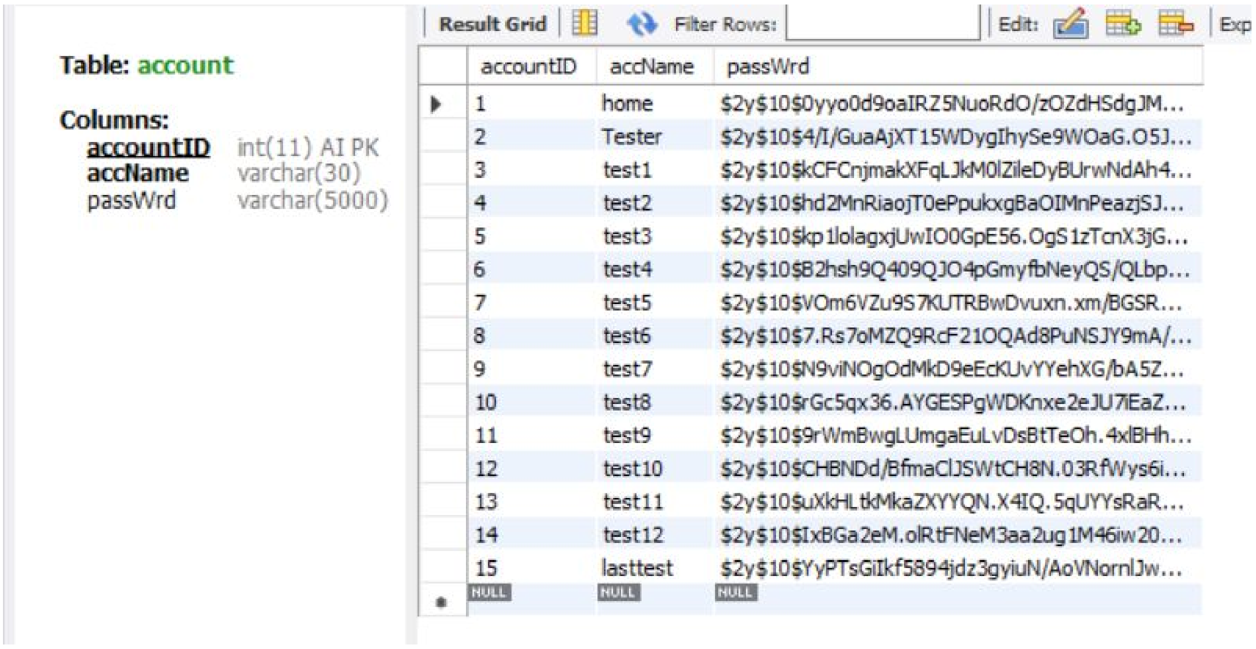


Table: account

Columns:

accountID	int(11) AI PK
accName	varchar(30)
passWrd	varchar(5000)

accountID	accName	passWrd
1	home	$2y$10$0yyo0d9oaIRZ5NuoRdO/zOZdHSdgJM...
2	Tester	$2y$10$4/I/GuaAjXT15WDygIhySe9WOaG.O5J...
3	test1	$2y$10$kCFCnjmakXFqLJkM0lZileDyBUrwNdAh4...
4	test2	$2y$10$hd2MnRiaojT0ePpukxgBaOIMnPeazjSJ...
5	test3	$2y$10$kp1lolagxjUwIO0GpE56.OgS1zTcnX3jG...
6	test4	$2y$10$B2hsh9Q409QJO4pGmyfbNeyQS/QLbp...
7	test5	$2y$10$VOm6VZu9S7KUTRBwDvuxn.xm/BGSR...
8	test6	$2y$10$7.Rs7oMZQ9RcF21OQAd8PuNSJY9mA/...
9	test7	$2y$10$N9viNOgOdMkD9eEcKUvYYehXG/bA5Z...
10	test8	$2y$10$rGc5qx36.AYGESPgWDKnxe2eJU7iEaZ...
11	test9	$2y$10$9rWmBwgLUmgaEuLvDsBtTeOh.4xlBHh...
12	test10	$2y$10$CHBNDd/BfmaClJSWtCH8N.03RfWys6i...
13	test11	$2y$10$uXkHLtkMkaZXYYQN.X4IQ.5qUYYsRaR...
14	test12	$2y$10$IxBGa2eM.olRtFNeM3aa2ug1M46iw20...
15	lasttest	$2y$10$YyPTsGiIkf5894jdz3gyiuN/AoVNornlJw...
NULL	NULL	NULL

Has

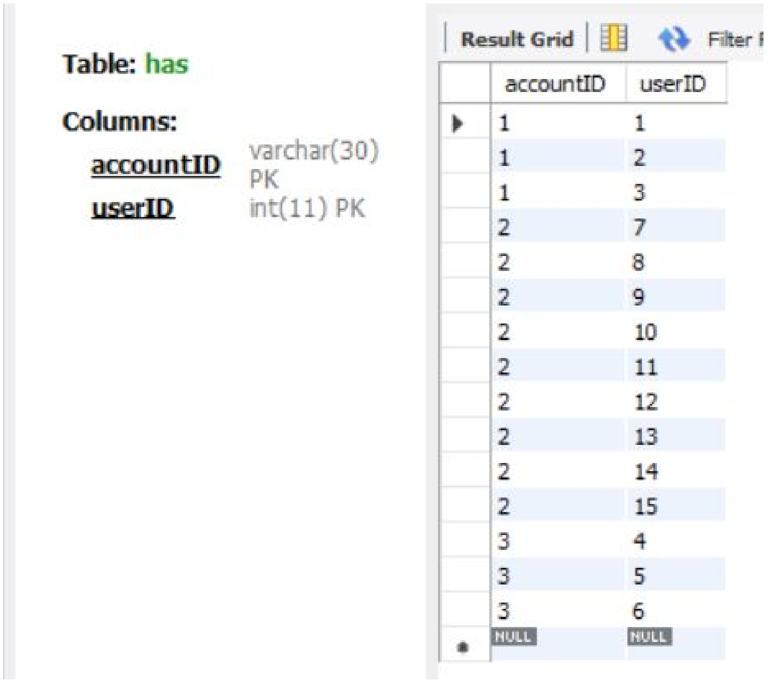


Table: has

Columns:

accountID	varchar(30) PK
userID	int(11) PK

accountID	userID
1	1
1	2
1	3
2	7
2	8
2	9
2	10
2	11
2	12
2	13
2	14
2	15
3	4
3	5
3	6
NULL	NULL

User

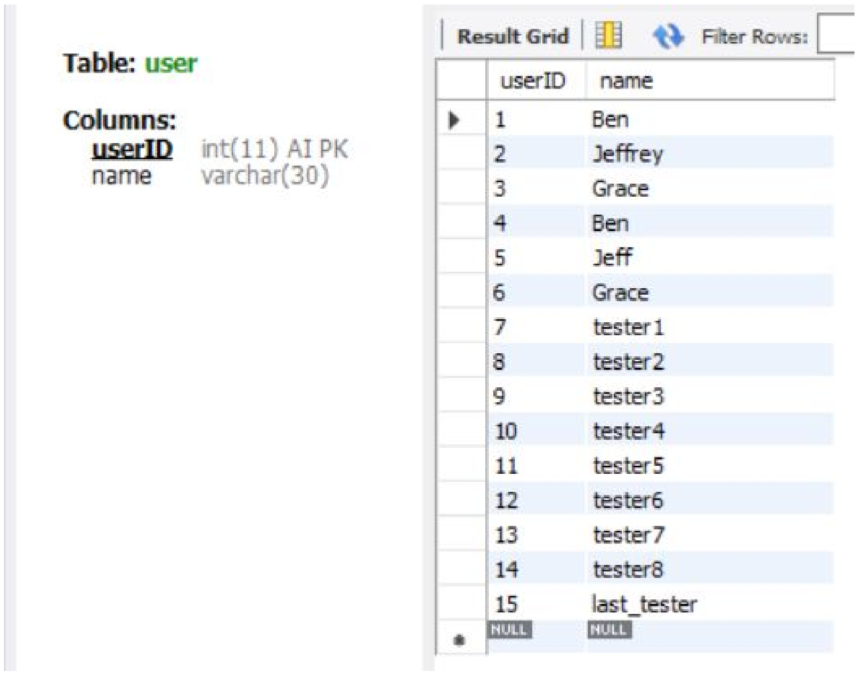


Table: user

Columns:

userID	int(11) AI PK
name	varchar(30)

userID	name
1	Ben
2	Jeffrey
3	Grace
4	Ben
5	Jeff
6	Grace
7	tester1
8	tester2
9	tester3
10	tester4
11	tester5
12	tester6
13	tester7
14	tester8
15	last_tester
NULL	NULL

Owns

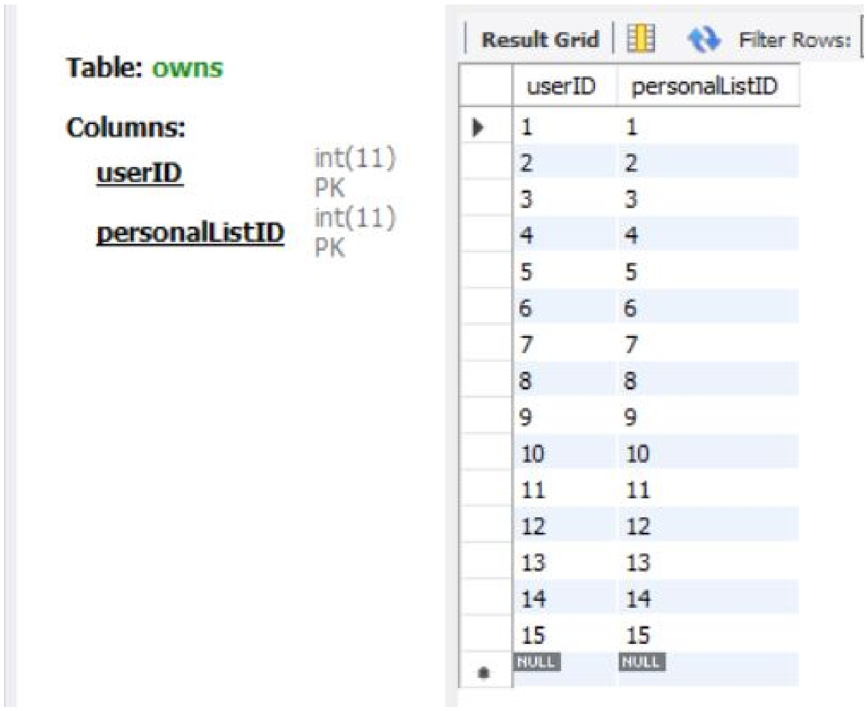


Table: owns

Columns:

userID	int(11) PK
personalListID	int(11) PK

userID	personalListID
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
NULL	NULL

PersonalList

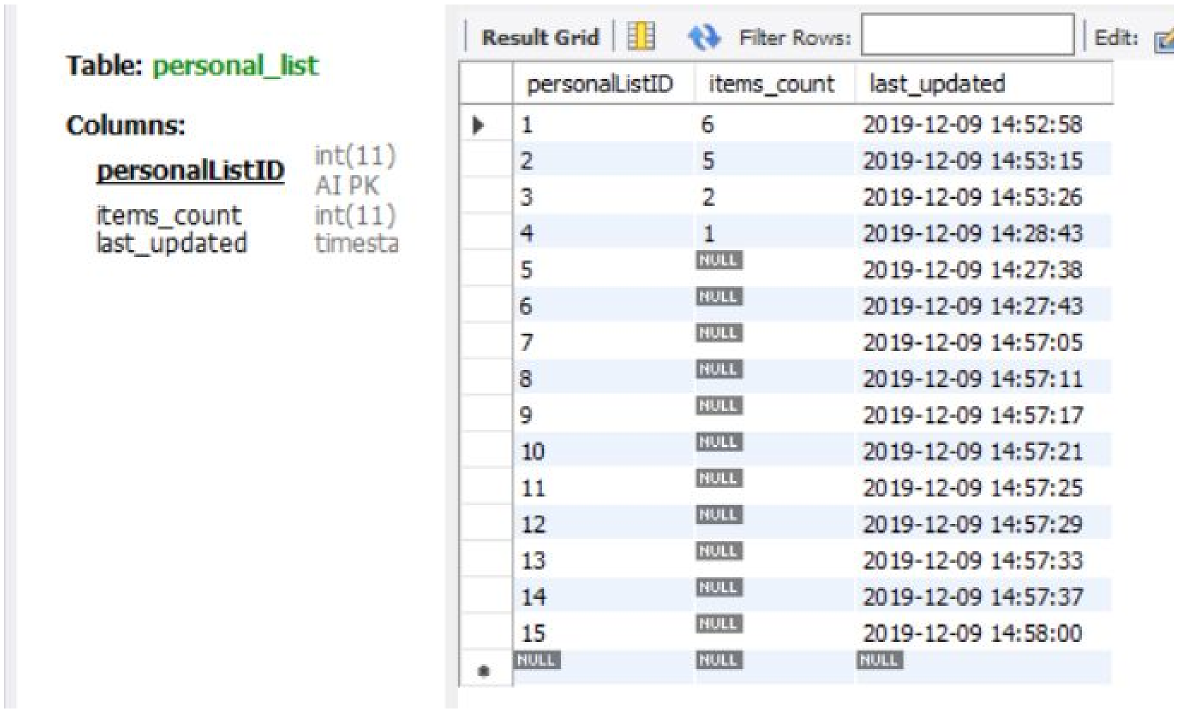


Table: personal_list

Columns:

personalListID	int(11) AI PK
items_count	int(11)
last_updated	timesta

personalListID	items_count	last_updated
1	6	2019-12-09 14:52:58
2	5	2019-12-09 14:53:15
3	2	2019-12-09 14:53:26
4	1	2019-12-09 14:28:43
5	NULL	2019-12-09 14:27:38
6	NULL	2019-12-09 14:27:43
7	NULL	2019-12-09 14:57:05
8	NULL	2019-12-09 14:57:11
9	NULL	2019-12-09 14:57:17
10	NULL	2019-12-09 14:57:21
11	NULL	2019-12-09 14:57:25
12	NULL	2019-12-09 14:57:29
13	NULL	2019-12-09 14:57:33
14	NULL	2019-12-09 14:57:37
15	NULL	2019-12-09 14:58:00
NULL	NULL	NULL

Contains

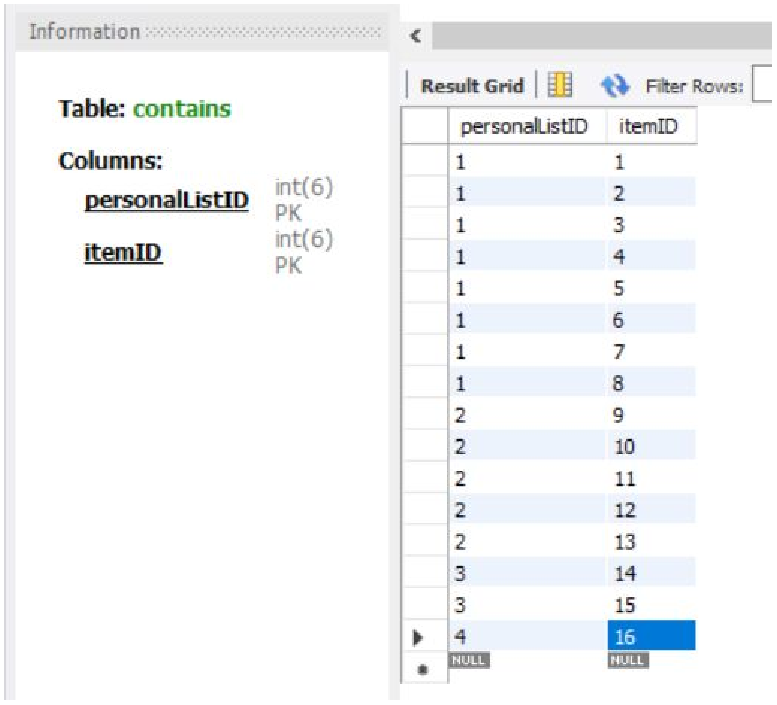


Table: contains

Columns:

personalListID	int(6) PK
itemID	int(6) PK

personalListID	itemID
1	1
1	2
1	3
1	4
1	5
1	6
1	7
1	8
2	9
2	10
2	11
2	12
2	13
3	14
3	15
4	16
NULL	NULL

Item

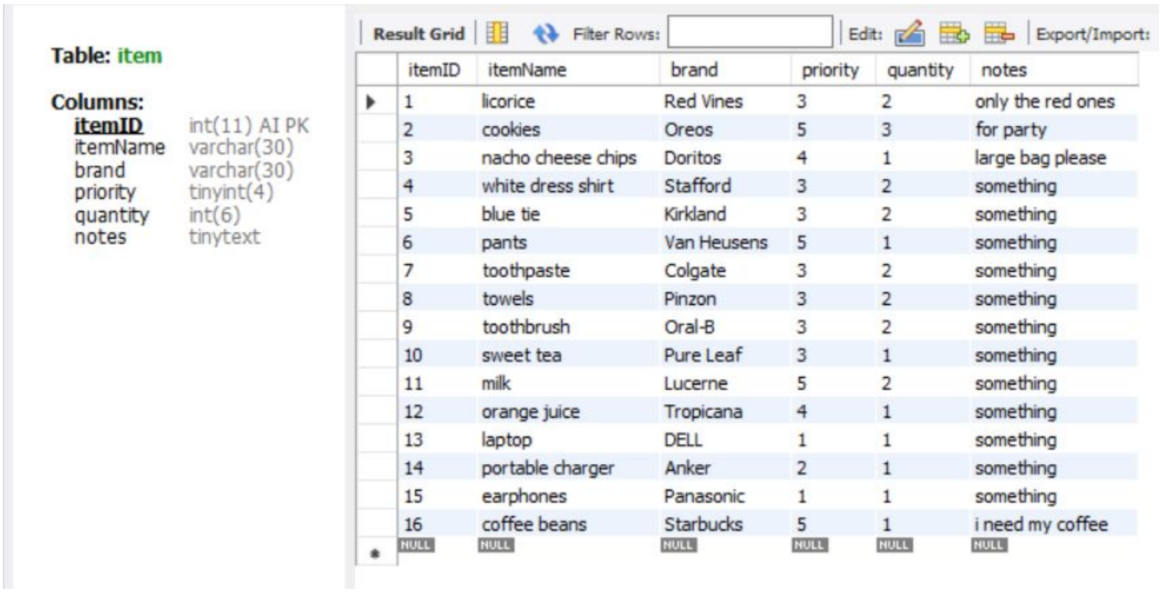


Table: item

Columns:

itemID	int(11) AI PK
itemName	varchar(30)
brand	varchar(30)
priority	tinyint(4)
quantity	int(6)
notes	tinytext

itemID	itemName	brand	priority	quantity	notes
1	licorice	Red Vines	3	2	only the red ones
2	cookies	Oreos	5	3	for party
3	nacho cheese chips	Doritos	4	1	large bag please
4	white dress shirt	Stafford	3	2	something
5	blue tie	Kirkland	3	2	something
6	pants	Van Heusens	5	1	something
7	toothpaste	Colgate	3	2	something
8	towels	Pinzon	3	2	something
9	toothbrush	Oral-B	3	2	something
10	sweet tea	Pure Leaf	3	1	something
11	milk	Lucerne	5	2	something
12	orange juice	Tropicana	4	1	something
13	laptop	DELL	1	1	something
14	portable charger	Anker	2	1	something
15	earphones	Panasonic	1	1	something
16	coffee beans	Starbucks	5	1	i need my coffee
NULL	NULL	NULL	NULL	NULL	NULL

Records

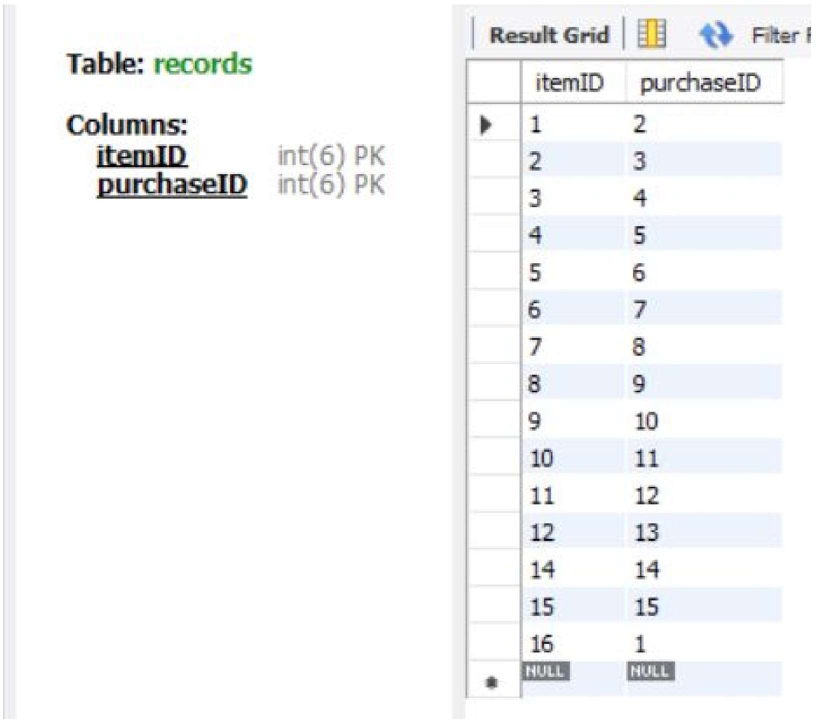


Table: records

Columns:

itemID	int(6) PK
purchaseID	int(6) PK

itemID	purchaseID
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	10
10	11
11	12
12	13
14	14
15	15
16	1
NULL	NULL

PurchaseHistory

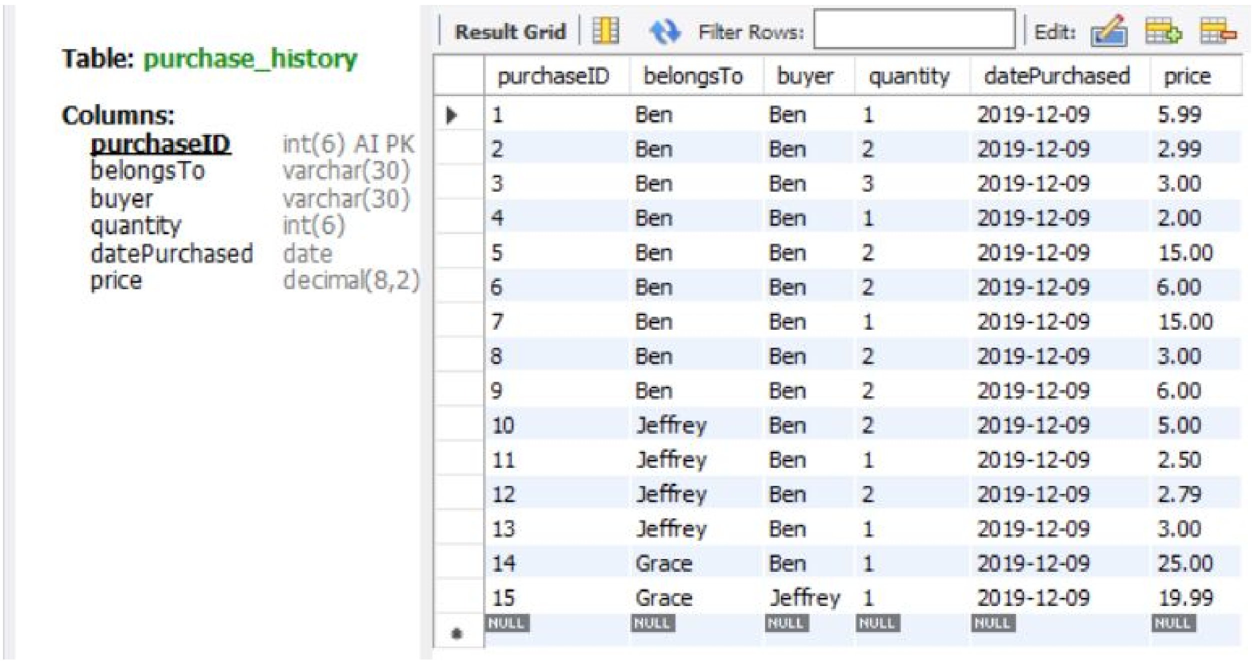


Table: purchase_history

Columns:

purchaseID	int(6) AI PK
belongsTo	varchar(30)
buyer	varchar(30)
quantity	int(6)
datePurchased	date
price	decimal(8,2)

purchaseID	belongsTo	buyer	quantity	datePurchased	price
1	Ben	Ben	1	2019-12-09	5.99
2	Ben	Ben	2	2019-12-09	2.99
3	Ben	Ben	3	2019-12-09	3.00
4	Ben	Ben	1	2019-12-09	2.00
5	Ben	Ben	2	2019-12-09	15.00
6	Ben	Ben	2	2019-12-09	6.00
7	Ben	Ben	1	2019-12-09	15.00
8	Ben	Ben	2	2019-12-09	3.00
9	Ben	Ben	2	2019-12-09	6.00
10	Jeffrey	Ben	2	2019-12-09	5.00
11	Jeffrey	Ben	1	2019-12-09	2.50
12	Jeffrey	Ben	2	2019-12-09	2.79
13	Jeffrey	Ben	1	2019-12-09	3.00
14	Grace	Ben	1	2019-12-09	25.00
15	Grace	Jeffrey	1	2019-12-09	19.99
NULL	NULL	NULL	NULL	NULL	NULL

Shows

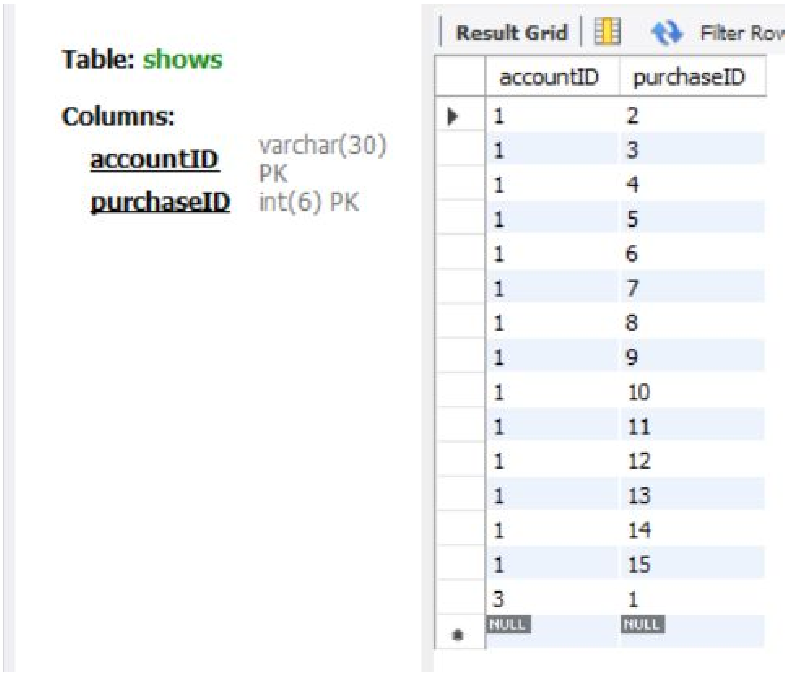


Table: shows

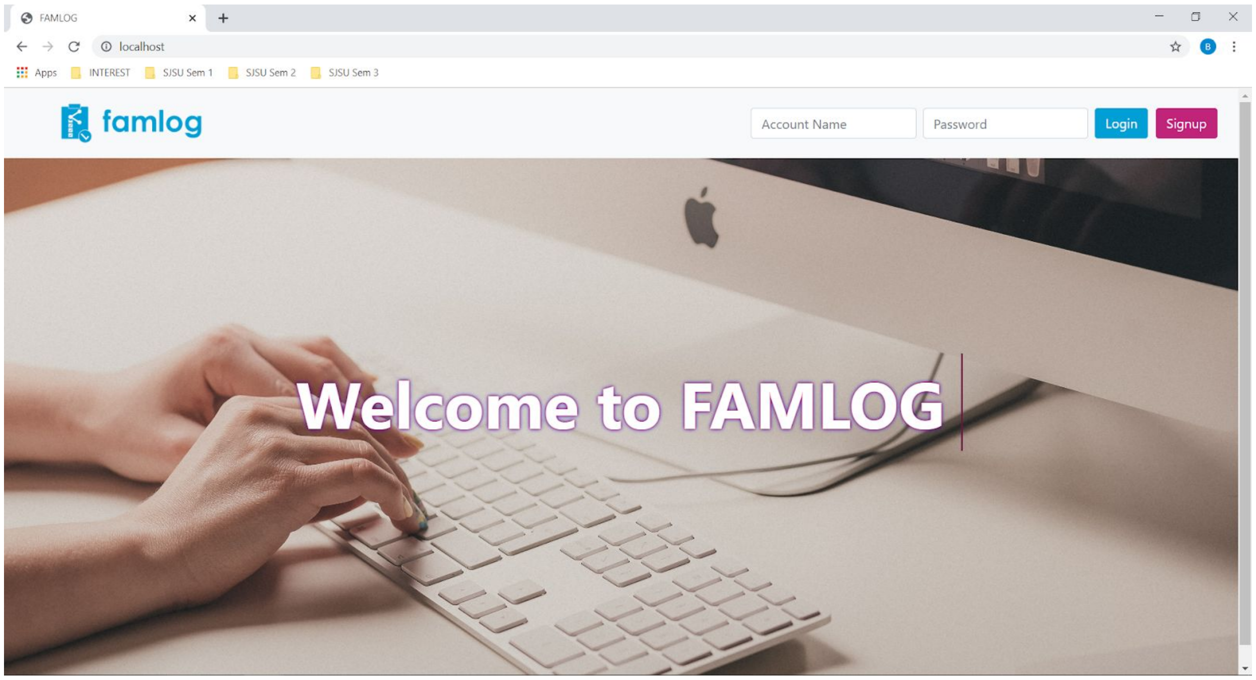
Columns:

accountID	varchar(30) PK
purchaseID	int(6) PK

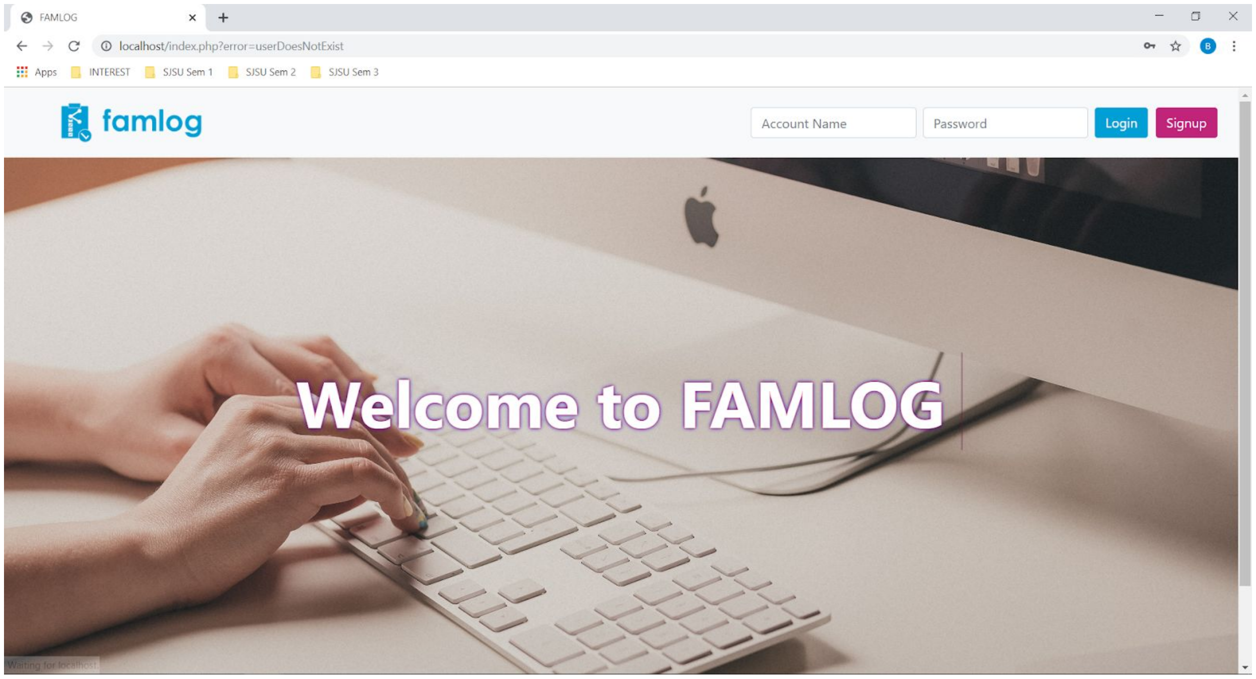
accountID	purchaseID
1	2
1	3
1	4
1	5
1	6
1	7
1	8
1	9
1	10
1	11
1	12
1	13
1	14
1	15
3	1
NULL	NULL

Implementation

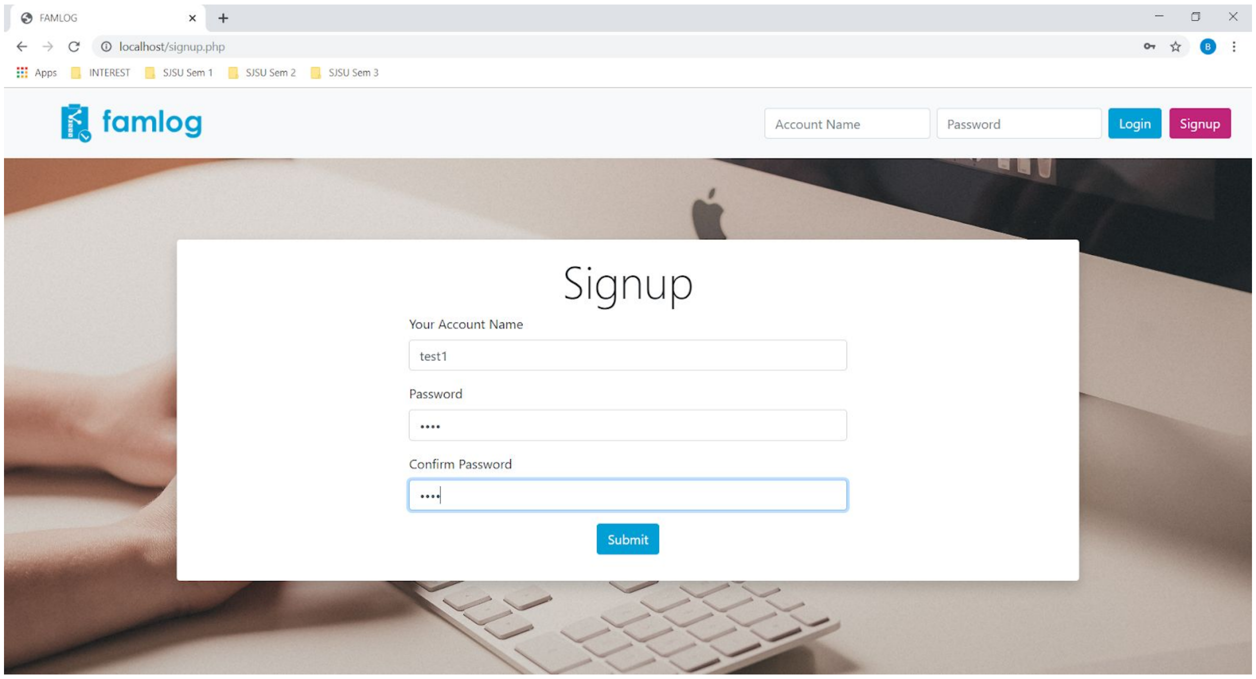
Landing page



Wrong login credentials (shown in the address bar)



User signup page



Signup

Your Account Name

test1

Password

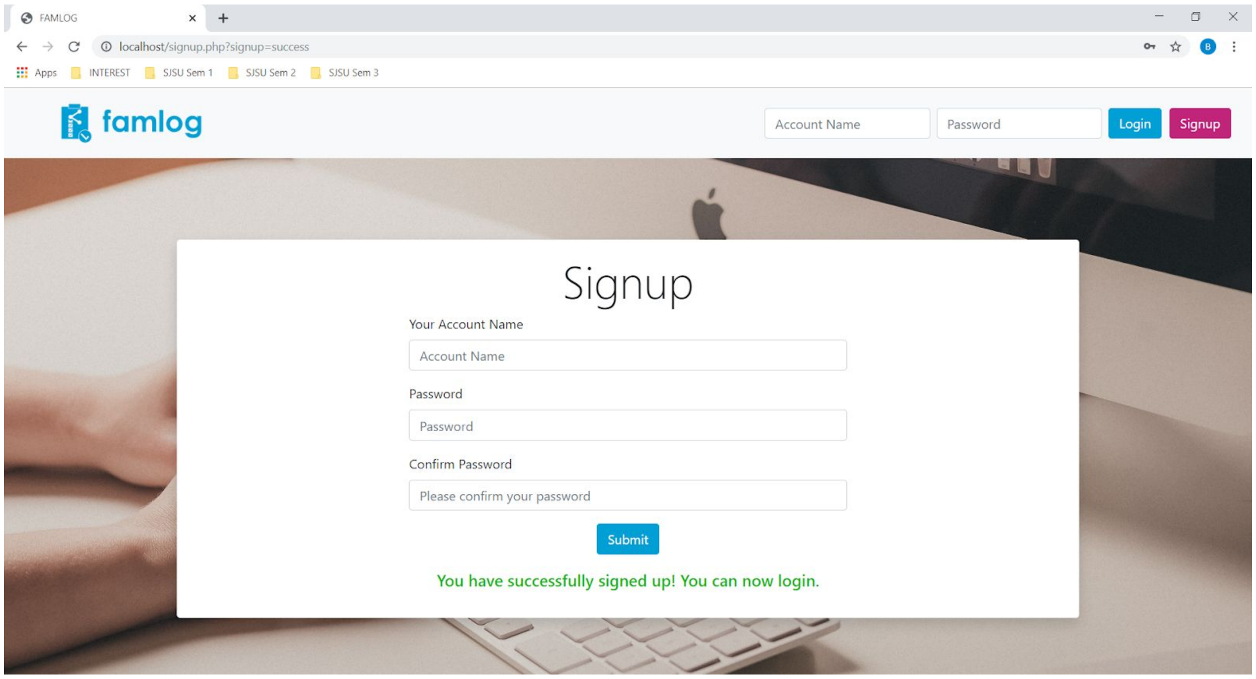
••••

Confirm Password

••••

Submit

Signup success



Signup

Your Account Name

Account Name

Password

Password

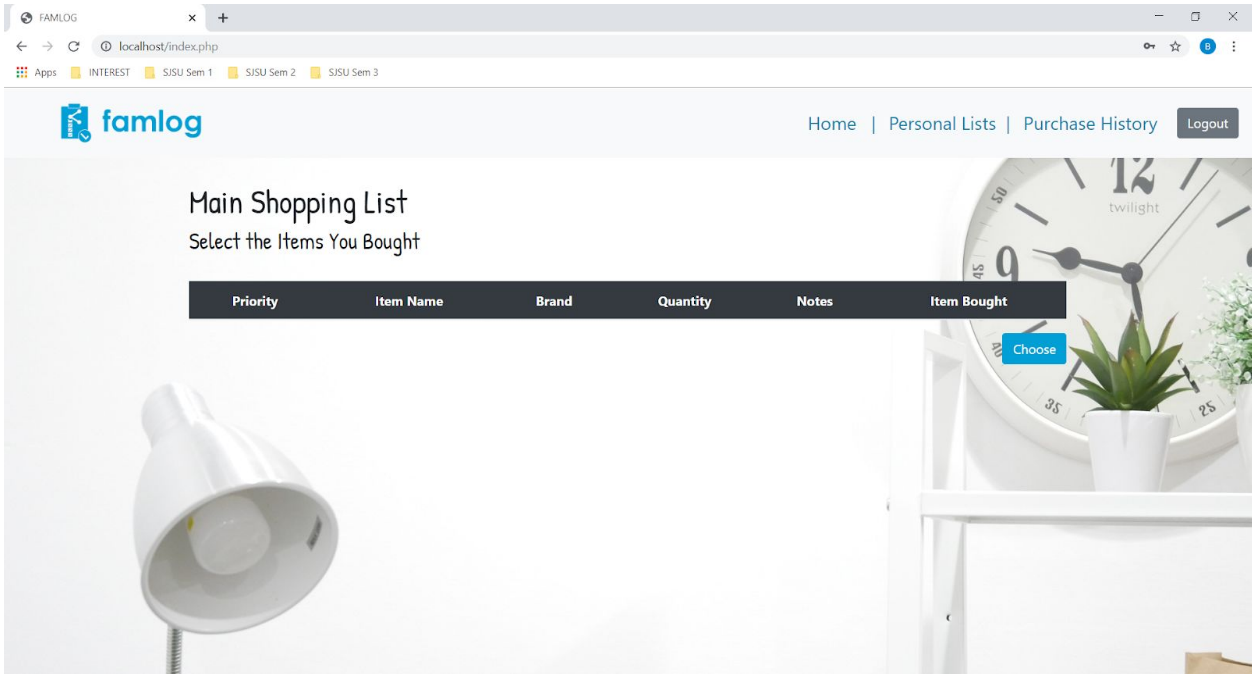
Confirm Password

Please confirm your password

Submit

You have successfully signed up! You can now login.

Account homepage (masterlist)

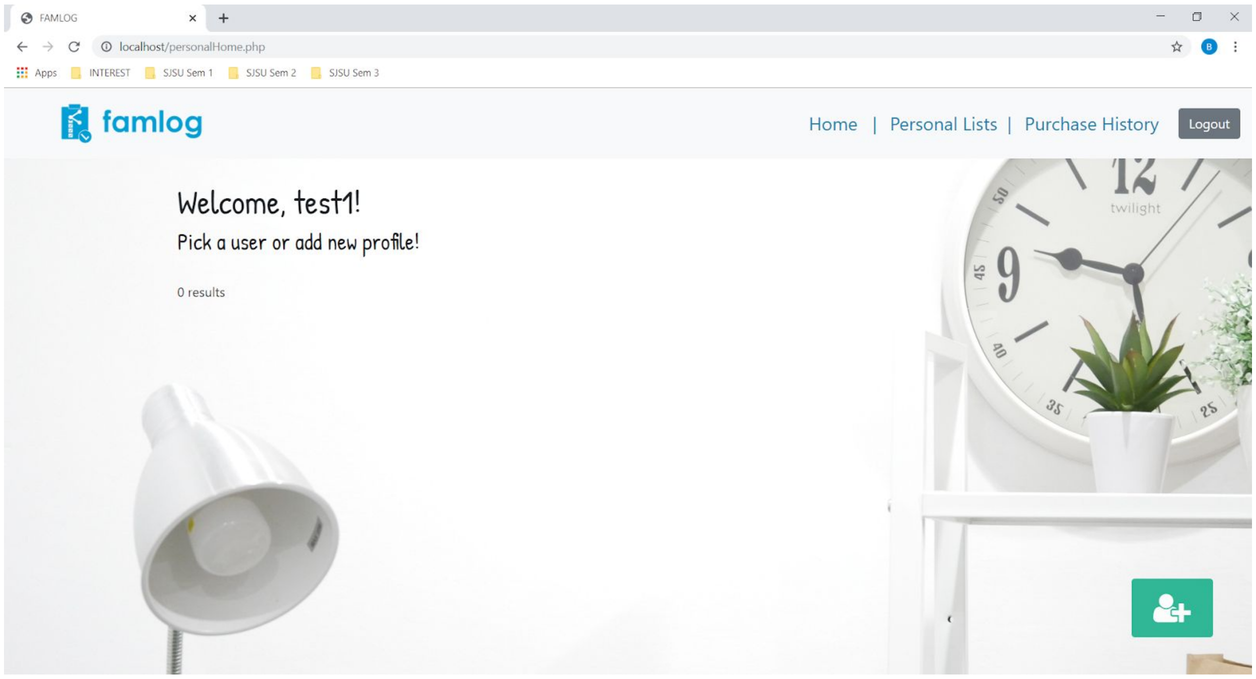


Main Shopping List

Select the Items You Bought

Priority	Item Name	Brand	Quantity	Notes	Item Bought

Account users page

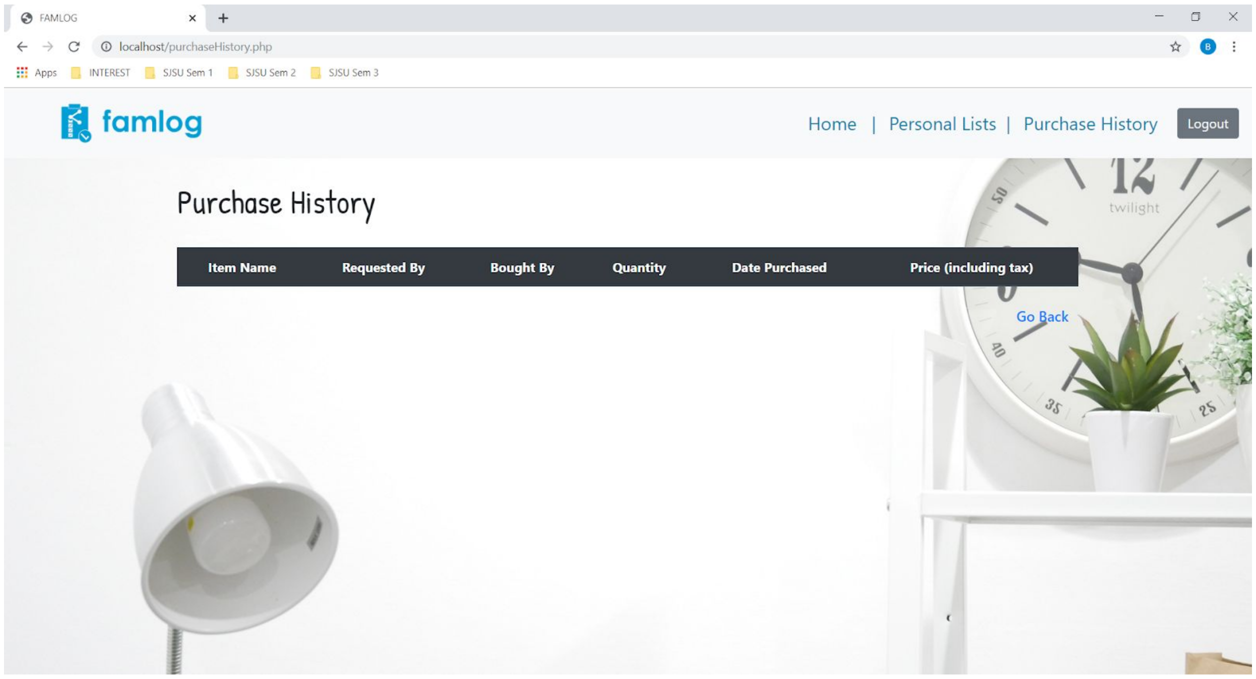

FAMLOG
localhost/personalHome.php
Apps
INTEREST
SJSU Sem 1
SJSU Sem 2
SJSU Sem 3
famlog
Home | Personal Lists | Purchase History
Logout

Welcome, test1!

Pick a user or add new profile!

0 results

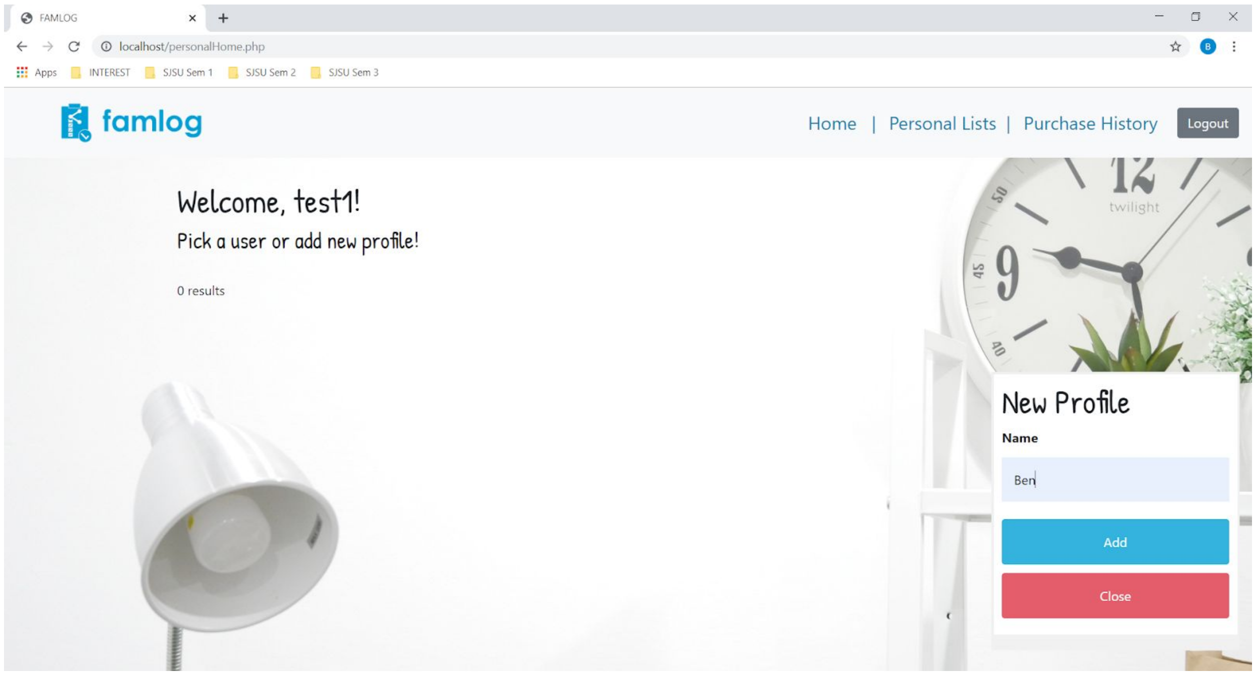
Account purchase history


FAMLOG
localhost/purchaseHistory.php
Apps
INTEREST
SJSU Sem 1
SJSU Sem 2
SJSU Sem 3
famlog
Home | Personal Lists | Purchase History
Logout

Purchase History

Item Name	Requested By	Bought By	Quantity	Date Purchased	Price (including tax)

Adding a new user


FAMLOG
localhost/personalHome.php
Apps
INTEREST
SJSU Sem 1
SJSU Sem 2
SJSU Sem 3
famlog
Home | Personal Lists | Purchase History
Logout

Welcome, test1!

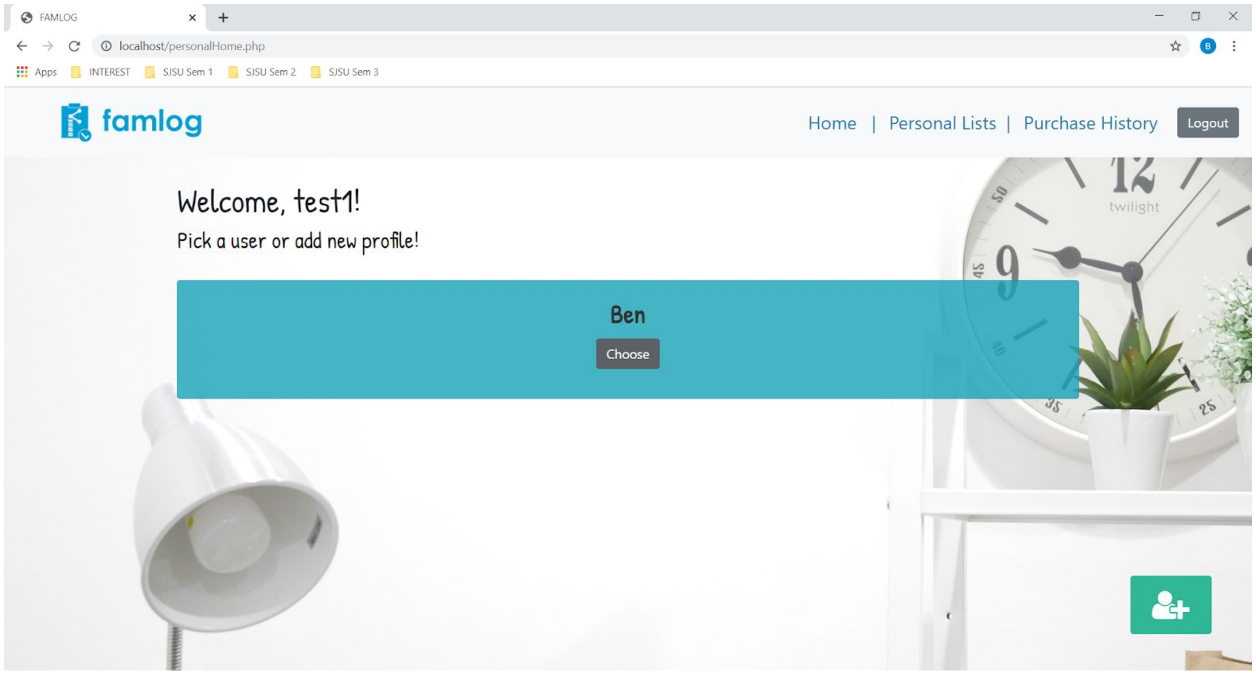
Pick a user or add new profile!

0 results

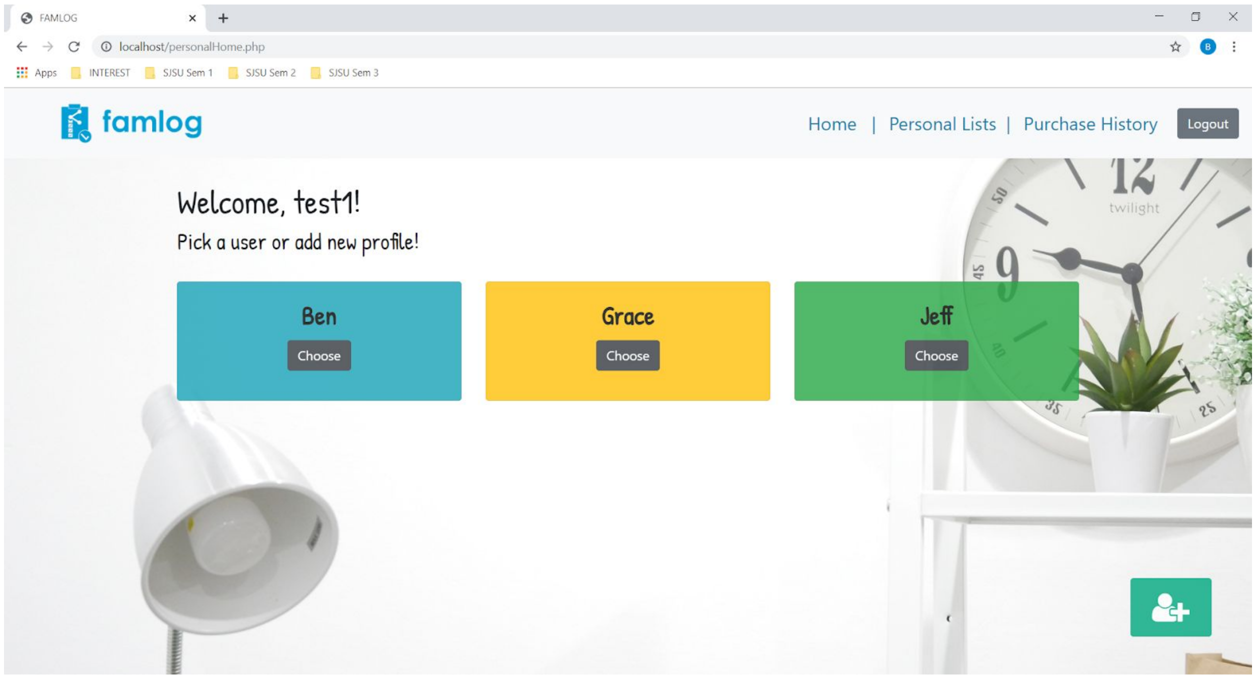
New Profile

Name: Ben

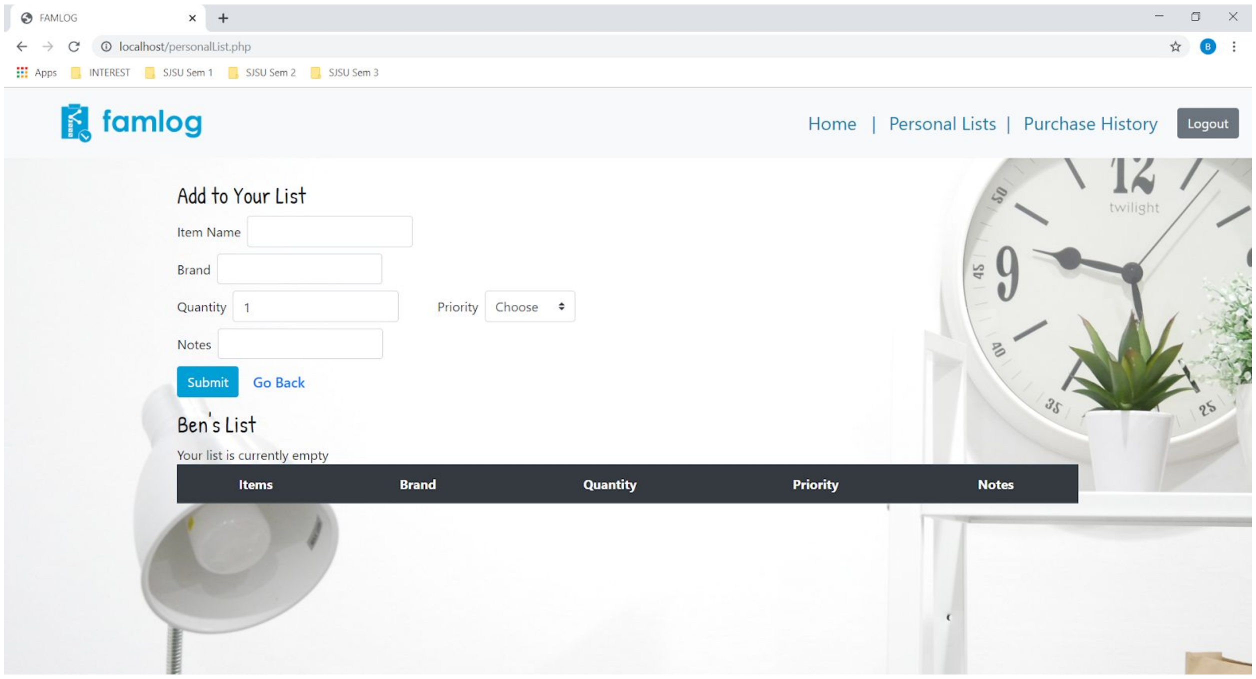
New user successfully added


FAMLOG
localhost/personalHome.php
Apps
INTEREST
SJSU Sem 1
SJSU Sem 2
SJSU Sem 3
famlog
Home | Personal Lists | Purchase History
Logout
Welcome, test1!
Pick a user or add new profile!
Ben
Choose

Added more users



Personal list for user - Ben


FAMLOG

localhost/personalList.php

Apps | INTEREST | SJSU Sem 1 | SJSU Sem 2 | SJSU Sem 3

famlog

Home | Personal Lists | Purchase History | Logout

Add to Your List

Item Name
Brand
Quantity 1
Priority Choose
Notes

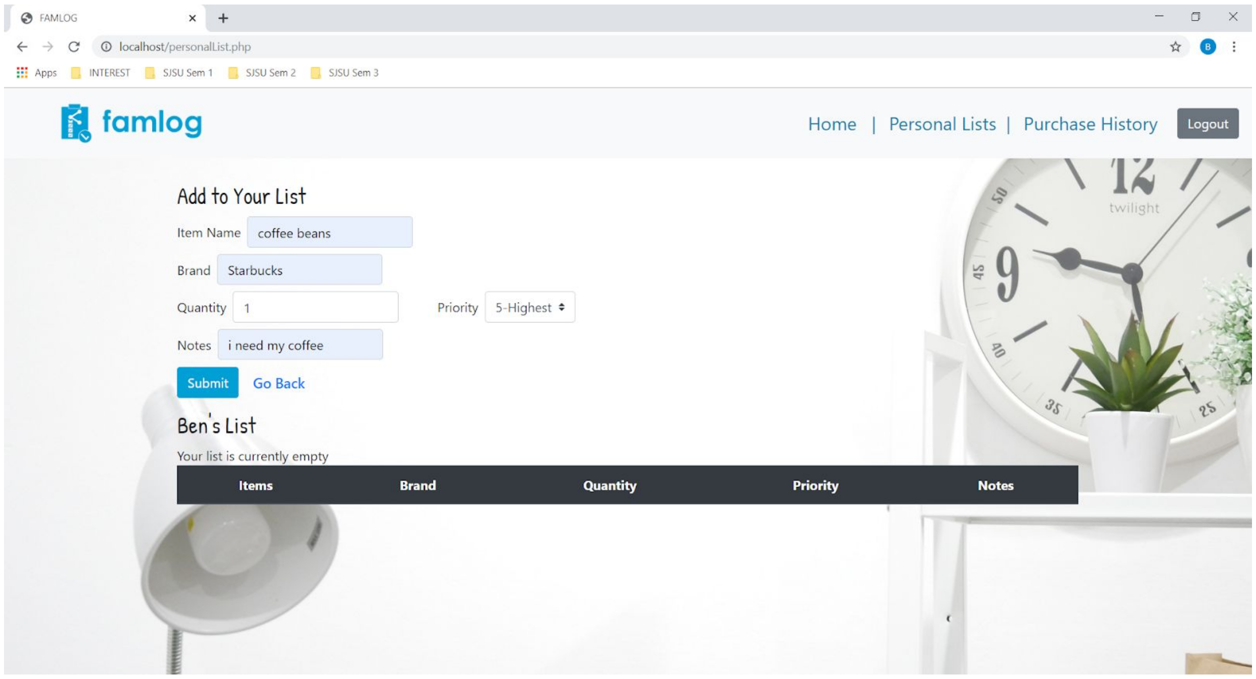
Submit | Go Back

Ben's List

Your list is currently empty

Items	Brand	Quantity	Priority	Notes

Adding an item to Ben's personal list


FAMLOG

localhost/personalList.php

Apps | INTEREST | SJSU Sem 1 | SJSU Sem 2 | SJSU Sem 3

famlog

Home | Personal Lists | Purchase History | Logout

Add to Your List

Item Name coffee beans
Brand Starbucks
Quantity 1
Priority 5-Highest
Notes i need my coffee

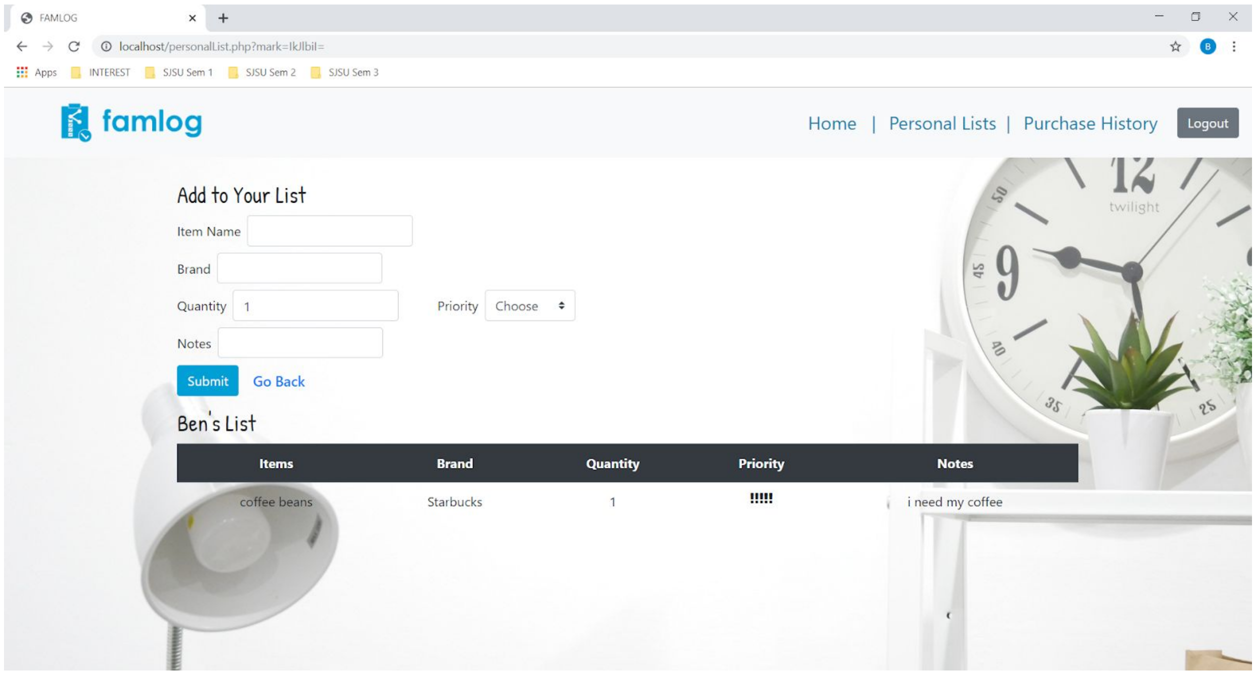
Submit | Go Back

Ben's List

Your list is currently empty

Items	Brand	Quantity	Priority	Notes

Item successfully added


FAMLOG
localhost/personalList.php?mark=lkJlbil=
Apps
INTEREST
SJSU Sem 1
SJSU Sem 2
SJSU Sem 3
famlog
Home | Personal Lists | Purchase History
Logout

Add to Your List

Item Name

Brand

Quantity 1

Priority Choose

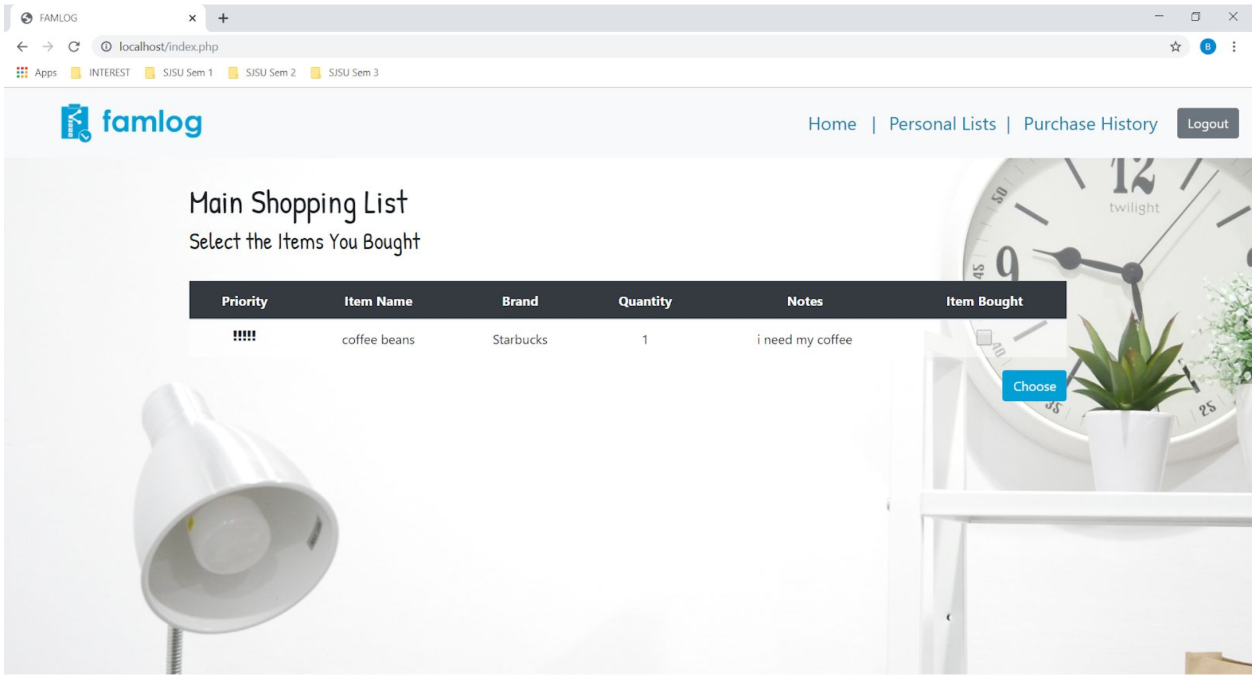
Notes

Submit Go Back

Ben's List

Items	Brand	Quantity	Priority	Notes
coffee beans	Starbucks	1	!!!!!	i need my coffee

Added item shown in master list



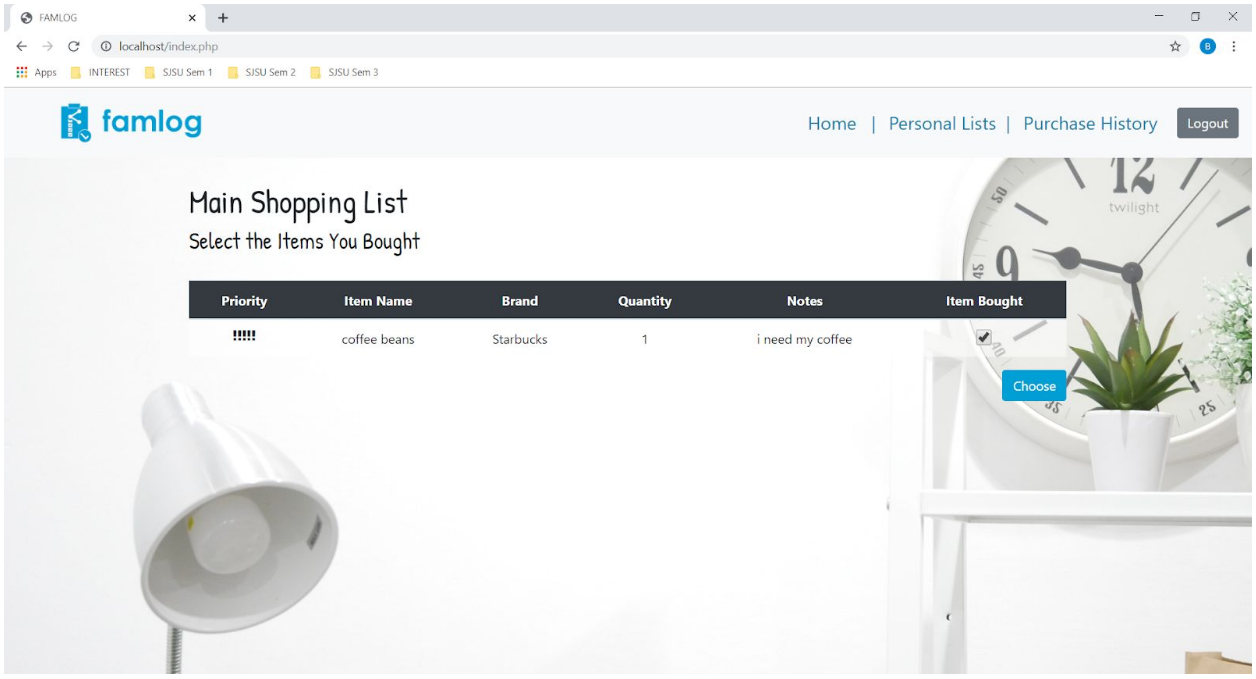
Main Shopping List

Select the Items You Bought

Priority	Item Name	Brand	Quantity	Notes	Item Bought
!!!!!	coffee beans	Starbucks	1	i need my coffee	☐

Choose

Item can be selected

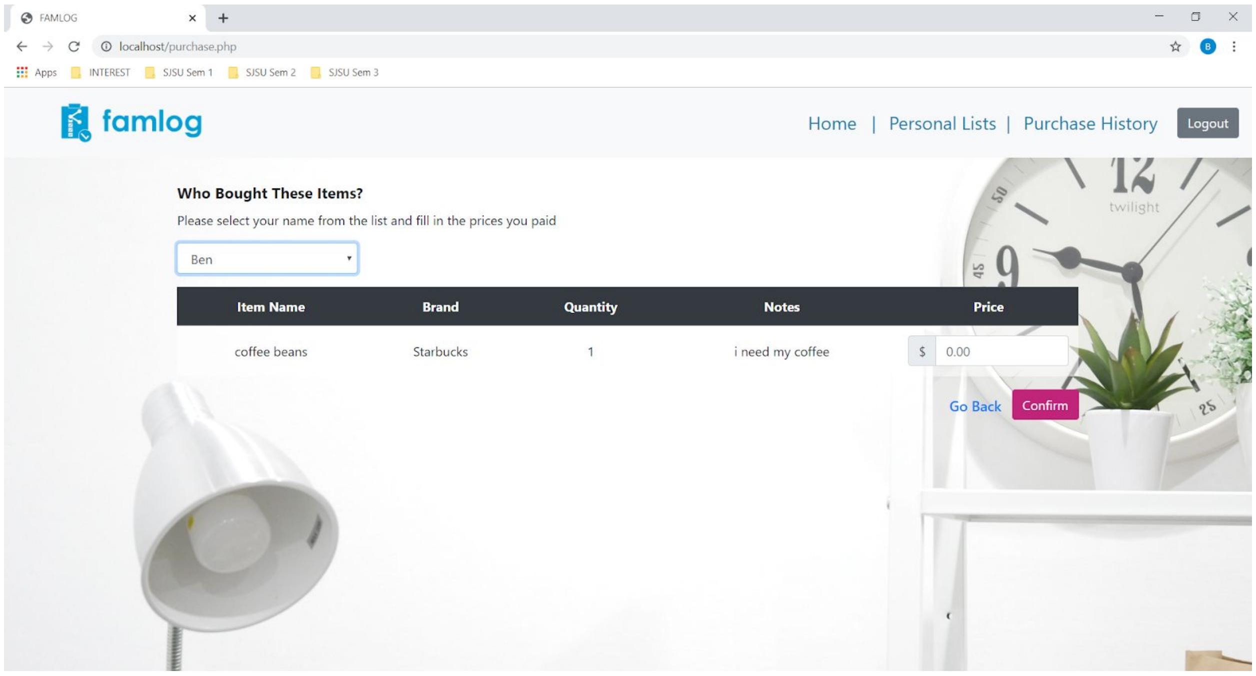


Main Shopping List

Select the Items You Bought

Priority	Item Name	Brand	Quantity	Notes	Item Bought
!!!!!	coffee beans	Starbucks	1	i need my coffee	✓

The user who purchases the item can choose his/her name



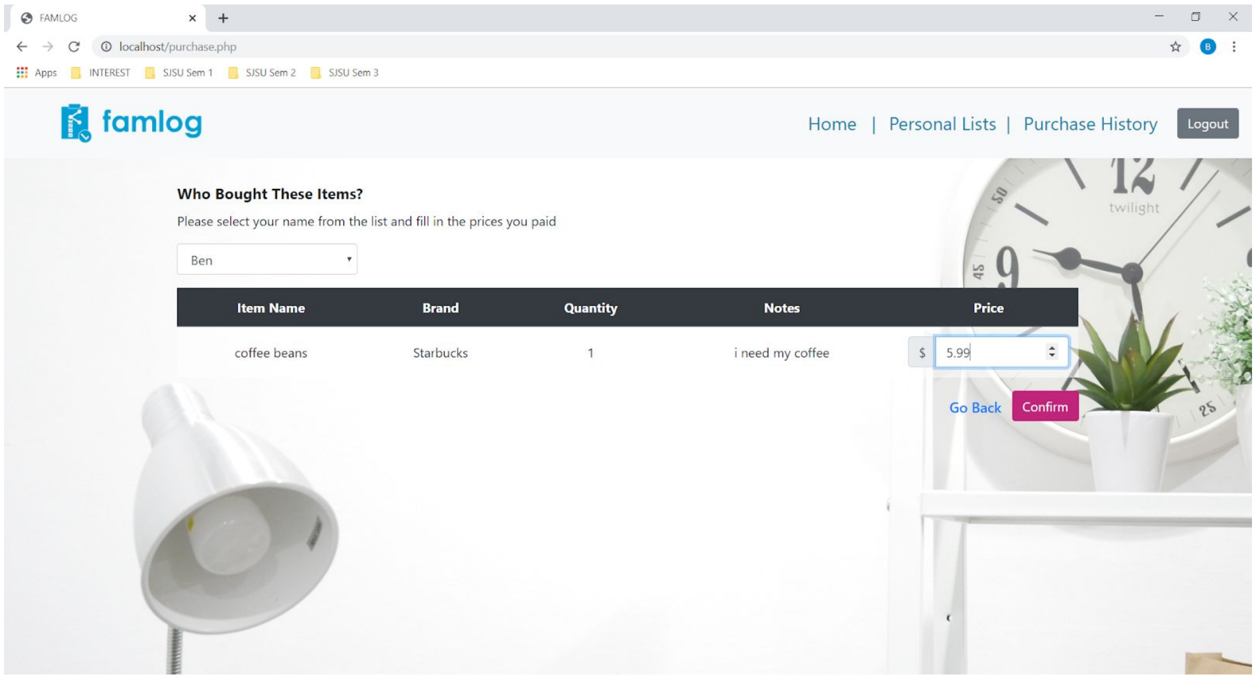
Who Bought These Items?

Please select your name from the list and fill in the prices you paid

Ben

Item Name	Brand	Quantity	Notes	Price
coffee beans	Starbucks	1	i need my coffee	$ 0.00

The buyer of the item can input the price



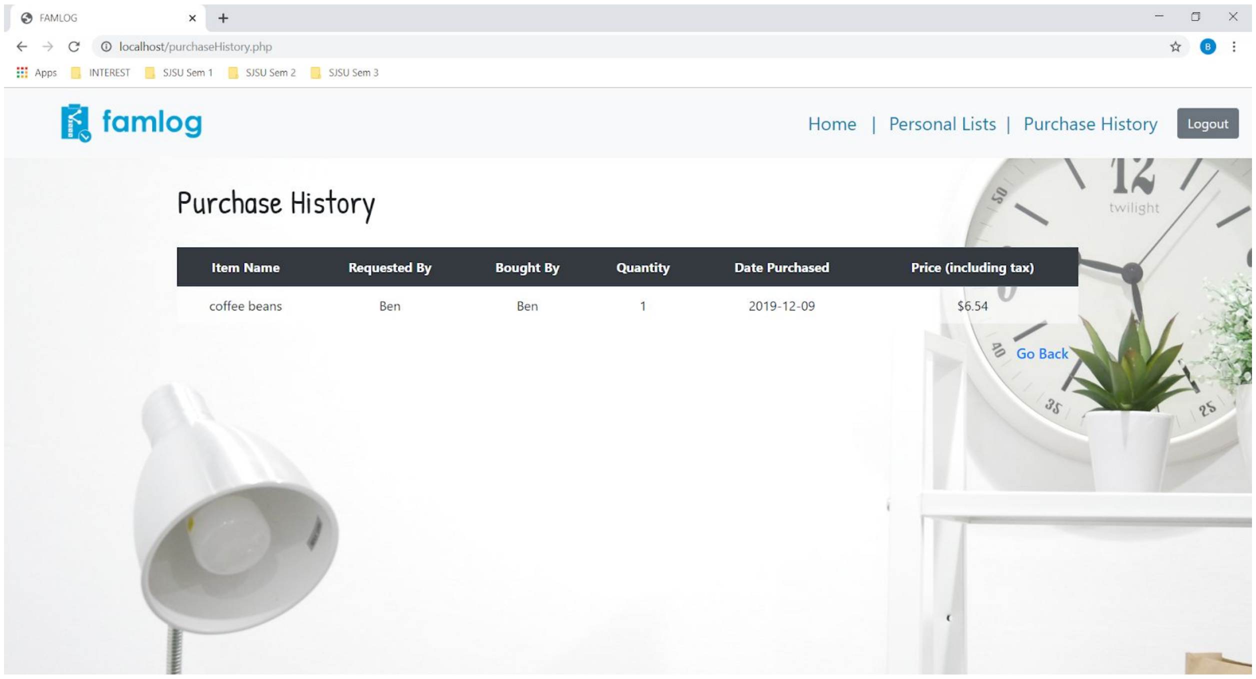
Who Bought These Items?

Please select your name from the list and fill in the prices you paid

Ben

Item Name	Brand	Quantity	Notes	Price
coffee beans	Starbucks	1	i need my coffee	$ 5.99

The item will appear in the purchase history, with the price calculated including 9.25% tax.



FAMLOG
localhost/purchaseHistory.php
Apps
INTEREST
SJSU Sem 1
SJSU Sem 2
SJSU Sem 3
famlog
Home | Personal Lists | Purchase History
Logout

Purchase History

Item Name	Requested By	Bought By	Quantity	Date Purchased	Price (including tax)
coffee beans	Ben	Ben	1	2019-12-09	$6.54

How to run:

1. Clone 'famlog' repository and place it in 'C:\Apache24\htdocs' (extract 'famlog' contents into 'htdocs', instead of having the entire 'famlog' folder in 'htdocs')
2. Create a schema in MySQL called 'cs157a_project'
3. Open a new query tab and paste the following:

```
USE cs157a_project;

CREATE TABLE account ( accountID INT NOT NULL AUTO_INCREMENT PRIMARY
KEY, accName varchar(30) NOT NULL UNIQUE, passWrd varchar(5000) NOT NULL )
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE user ( userID INT NOT NULL AUTO_INCREMENT, name varchar(30)
NOT NULL, PRIMARY KEY (userID) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE has ( accountID varchar(30) NOT NULL, userID INT NOT NULL,
PRIMARY KEY (accountID,userID) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE item ( itemID INT NOT NULL AUTO_INCREMENT, itemName
varchar(30) NOT NULL, brand varchar(30) DEFAULT NULL, priority TINYINT NOT NULL,
quantity int(6) NOT NULL, notes tinytext, PRIMARY KEY (itemID) ) ENGINE=InnoDB
DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE owns ( userID INT NOT NULL, personalListID INT NOT NULL,
PRIMARY KEY (userID,personalListID) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE TABLE personal_list ( personalListID INT NOT NULL AUTO_INCREMENT,
items_count INT, last_updated timestamp default current_timestamp on update
current_timestamp, PRIMARY KEY (personalListID) ) ENGINE=InnoDB DEFAULT
CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE TABLE purchase_history ( purchaseID int(6) NOT NULL AUTO_INCREMENT,
belongsTo varchar(30) NOT NULL, buyer varchar(30) NOT NULL, quantity int(6) NOT NULL,
datePurchased date NOT NULL, price decimal(8,2) NOT NULL, PRIMARY KEY (purchaseID)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE TABLE records ( itemID int(6) NOT NULL, purchaseID int(6) NOT NULL,
PRIMARY KEY (itemID,purchaseID) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE TABLE shows ( accountID varchar(30) NOT NULL, purchaseID int(6) NOT NULL,
PRIMARY KEY (accountID,purchaseID) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE TABLE contains ( personalListID int(6) NOT NULL, itemID int(6) NOT NULL,
PRIMARY KEY (personalListID,itemID) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

4. Run these queries
5. Open web browser and enter 'localhost' in the address bar
6. You are all set!

Project Conclusion

Through this project, we learned how to fully build a website from nothing. From brainstorming a project idea, selecting the tools, sketching the ER diagram, listing out the requirements and finally, implementation. Although some of us have some knowledge in PHP, for all of us, this was the first time having to make a website from scratch, but we still managed to pull through. We learned how important it was to make sure that our ER diagram is accurate, so that our tables in the database will be correct. We also learned how to set up connections to MySQL and perform queries. For the User Interface, it was interesting to learn how to implement buttons which links to other pages, make selections interactive (allow users to choose between 1-5 item priority), and also submission forms which stores data in the local database. We also learned how to hash a password for added security.

If we were given a chance to redo the project, we would improve in our planning and implementation processes. We switched from PHP to NodeJS back to PHP, due to time and ability constraints. Because of this, we also started our project late, because we did not have enough time to learn a new language. For future projects, we would stick with languages which the team is familiar with.

We believe that we have achieved the basic functions of our program, but there is still room for improvement in the security aspect and also functionality. For future improvements, we could add more functions, such as edit buttons and delete buttons. We could also make the price display in total for each transaction, other than listing it one by one. Our program is self-explanatory and easy to navigate, so we would focus more on functionality in the future.