

# Web Application Development Project

## Fine Foods by James Taylor G00012318

**Installation guide:** This website was designed and tested on chrome Version 54.0.2840.99 m

- 1) For this website to work you need to first have XAMPP installed on your windows machine. Go to <https://www.apachefriends.org/download.html> and follow the instructions. XAMPP is an easy to install Apache distribution containing MariaDB, PHP, and Perl. Just download and start the installer. It's that easy.
- 2) Once XAMPP is installed open the XAMPP control Panel and start-up APACHE and MYSQL. The localhost web server and MariaDB will not work until you do this.
- 3) Install the fine\_foods database.
  - a) Open the Command Prompt on your windows machine. Go to the mysql bin directory. Default is usually "C:\xampp\mysql\bin"
  - b) Enter the following and hit enter: `mysql -h localhost -u root -p`  
Password is not setup by default. So leave blank or enter your pw if you setup one and hit enter.
  - c) Once connected run the following commands to set up the database and its tables hitting enter after each one.
    - 1: `create database fine_foods;`
    - 2: `use fine_foods;`The command line should show you are currently in the fine\_foods database as follows:  
"MariaDB [fine\_foods]>"
    - 3: `create table users(  
    user_id int unsigned zerofill not null auto_increment primary key,  
    fname varchar(255),  
    lname varchar(255),  
    uname varchar(255),  
    pw varchar(255)  
);`  
You will see the following notifying you it has run ok "Query OK, 0 rows affected (0.53 sec)"
      - 4: `alter table users add unique key uname (uname);`
      - 5: `create table orders (  
    order_id int not null auto_increment primary key,  
    username varchar(255) not null,  
    order_total float not null  
);`
      - 6: Finally you can exit: `exit`

Congratulations you have now created the database for the website to use.

- 4) Download the files from GitHub in the correct location under a new folder to your local htdocs folder. If you installed xampp using the default settings this folder should be located in "C:\xampp\htdocs". **Note you need to place the files under a new folder into your local htdocs folder. Do not simply download the files to a 2nd htdocs folder inside your htdocs folder. Download all the files to a new folder and then move the folder to your htdocs folder.**

### Projects Optional requirements:

For styling the website I have three css files. One is located in the "includes" folder for the login, register, view\_users , validate and confirm pages. And for the home(index), lunch, dinner, dessert and sholist pages I used both Bootstrap "css/bootstrap.css" and another file "css/myStyles.css". This allows for a dynamic website with content flow and reposition for when the browser window size is changed. Curser pointer is used for buttons and links. A hide, delay and fade-in styles have been added for the item links and total price text. Comments have been added to myjs and cc\_validator in the JS folder to follow the code and understand where and what is it used for. Also the username was populated to top left of nav bar for user reference and as a link to view items ordered.

### Core Requirements:

- 1) Login/register page: "login.php" is the first page that a user is directed to in the project link in the module webpage. This includes a header.html file which offers links to register.php and view\_users.php pages. Each of the three includes the same header to navigate across. Passwords for users are stored using SHA1 for encryption (register.php line 26). mysqli\_connect.php is used to make the connection to the database and is called in login.php line 14. When login is a success a username cookie is created.
- 2) User is directed to home page once logged in or registered. If user chooses to go back to login screen he/she is logged out and orders are cleared. This is to stop any orders from one user been added to another users account.
- 3) Once user selects items from the ingredients lists in lunch/dinner/dessert html pages the items will be added to a localStorage object "orders" with name and price. A hide/delay/fade-in style was added for effect to show an item was added once clicked. Items count is recorded in a cart icon located at the top right of each page. (not in shopping list page as items are counted in a list). A user can navigate these pages through the nav bar in the top right using a dropdown list of the pages. Also to the left is the current username displayed with a link to the shopping cart.

- 4) Using the dropdown list in the nav bar or by clicking on his/her username to the top left a user can go to the shopping list page. Here the products/items are added to a cart/list. An event listener is used on this page to populate the items. (myjs.js line 74). Name, cost and total are populated by using a for loop (myjs.js line 89) of the localStorage object orders. Hide/delay/fade-in style was used for the total cost line at the bottom of the list. I used an Ordered List structure to display the ordered items.
- 5) Event listener to remove items (myjs.js line 108) was added at parent level of OL tag in shoplist.html line 62.
- 6) Using JS the event listener, using bubble flow (false), was used to target a row for deletion as it is clicked on. Also the orders object , cart count and the total localStorage variables are updated. (myjs lines 118 to 143)
- 7) The shoplist html page has a “Proceed to process order” button to bring the user to a new page to enter payment details. (line in HTML 65)
- 8) The validate html page invokes a plugin “js/jquery.creditCardValidator.js” to validate the credit card. The cc\_validator.js file is used here.
- 9) Once the user enters a valid credit card number and hits confirm this will write the order to the database by invoking the confirm.php script. Username and total are added to the orders table in the database “fine\_foods”. Confirm.php line 47 “\$orderId” is returned from the DB to get the auto incremented order\_id value and populates it to the screen for the users reference.

The connection is made by invoking the “mysqli\_connect.php” script. No password is set to the root access. \$dbc is then referred to in confirm.php to run the insert sql to the database. The total cookie is cleared once the order was sent to the DB to avoid user having duplicate orders by refreshing the page confirm.php. This cookie is checked at the start of the script. Also line 53 will call the function listed in the script html tags at lines 8 to 12 to clear orders localStorage so when user goes back to shoplist html the carts list will be empty.