Pizza Website Project

What to do:

For this project we will build a website for a pizza restaurant. We will start with a Minimum Viable Product (MVP) and then add on additional functionality as “the client” requests it. After step one feel free to skip ahead if you feel like you can do it without any of the previous steps.

This is a style of development where you guarantee to have something out even if it’s only a minimally useful project, then add on the features that increase it’s usefulness as time goes on. This way the client can cancel the project early if they run out of funding and still have a working product, or if the project takes longer than expected there will be a minimally functioning project in the worst case rather than a large project where everything half works.

Stages:

1. Build a website that can take a pizza order. It should have three pages:
    a. A page to fill out an order
    b. A page to show the customer their completed order
    c. A page where the owner can view all orders
2. Style the site so that it looks nice. If it would make you think, “this site seems cheap/sketchy” then keep styling it. Try and copy styles on sites you think look good (they don’t have to be pizza sites).
3. Update the view order page to show only orders that have not been fulfilled. Also add a button that allows the owner to mark orders as completed so they won’t show up any more. *This way the owner will only see orders that he/she needs to fulfill.*
4. Create a page that shows how many orders and how much money the restaurant has gotten over each of the last 7 days, 7 weeks, and 7 months. *This will allow the owner to try different specials and see if those specials increase revenue.*
5. Create a toppings popularity page that shows which toppings are the most popular. *This will allow the owner to figure out what his clients like best and offer specials around that.*
6. Allow the sale of specialty, pre-structured pizzas
    a. Add some forms to the top of the page with pre-made pizza types (cheese, pepperoni, hawaiian)
7. Allow discounted toppings: *This might get customers to add toppings they otherwise wouldn’t have*
    a. Create a page where the owner can discount toppings.
    b. Show the discount on the order page.
8. Create a page to manage (edit, add, and delete) the different types of specialty pizzas. *This way the owner can add new pizzas without asking you to do development work.*
9. Track users:
    a. Add a tracking cookie to the user’s browser to identify them.

            i. Make sure it never expires (or only does after a really long time).
            ii. The tracking cookies should be a random `hash()`
    b. Associate each order with a user’s tracking cookie.
    c. Add their user id to the order view page.

10. Add a recurring customer discount
    a. On the order page add a small discount for a customer that has placed another order within the last week.
    b. Add a tooltip on the discount section explaining to the user why they are getting the discount. *This should hopefully get people to be repeat customers.*

Stage 1

Languages: HTML, PHP, and MySQL

Build three pages:

1. A page where a user can place an order for an individual pizza.
2. A page that shows the user their order.
    a. This page should also save the order to the database.
3. A page with a table where all of the orders can be viewed.
    a. The orders should be ordered by creation date so the most recent orders are at the top.

Step 0 - Recommended Database Setup

You don’t have to use this, but it is a decent setup for the database. Feel free to do whatever you want if you feel you can do it simpler, better, etc.

Order

Id int; Primary Key
Name varchar;
Address varchar;
Phone varchar;

Topping

Id int; Primary Key
Name varchar;
Price float;

OrderTopping
OrderId int; Primary Key part 1
ToppingId int; Primary Key part 2

The topping table will have each topping in it once. Bell Pepper will be in there once. Pepperoni will be in there once. Regular Cheese will be in there once. Extra cheese will be in there once. ETC.

When you submit an order you will create an Order in the database with the name, address, and phone number. When you get your response from the database it will have an id on it. Then submit an OrderTopping with that order id and the topping id of each topping on the pizza in your order.

Step 1 - The Order Form Page

This page is almost entirely HTML.

The user should be able to order one pizza with the following toppings (feel free to add some):

- Cheese ($0.50) or Extra Cheese ($1.00)
- Sauce:
    - Pesto (FREE)
    - Cheese (FREE)
    - Tomato (FREE)
- Meat:
    - Pepperoni ($1.00)
    - Sausage ($0.80)
    - Ham ($1.50)
    - Bacon ($1.30)
    - Anchovies ($2.00)
- Fruits and Veggies:
    - Bell Peppers ($0.30)
    - Onions ($0.10)
    - Olives ($0.55)
    - Pineapple ($1.00)

The cheese and sauce should be selected using a radio button (you can only have regular or extra cheese, not both). The rest of the toppings should be selected using a checkbox. The default price of a pizza is $5.00.

The user should also submit their address, name, and phone number.

This first page (the order form page) should submit the data to the second page (the order review page) using a POST request.

Step 2 - The Order Review Page

This "page" is where a lot of your PHP code should be.

The purpose of this page is to actually save data and then be a receipt. You will want to do a few things here:

- Save the order that was submitted into the database.
- Print out the order details on the screen
- Print out the price.

Step 3 - The Order Fulfillment Page

This page's purpose is to be viewed by the pizza store owner to see what orders they need to make.

This page should list out ALL of the orders along with the name, address, and phone number, so that the store owner can make those pizzas and then deliver them. No need to show prices on this page.

The most recently ordered pizza should be at the top. It's not useful if they always see the first pizza ever ordered at the top. Try using the "ORDER BY" syntax in your "SELECT" statement.

Stage 2

Languages: HTML and CSS

There are no specific instructions for this step. Look at your favorite online stores: Amazon, Walmart, Etsy, another pizza restaurant, some shoe store, or pet store (doesn't matter) and find styles for design. Try and mimic things that you like and professionals are doing.

Make sure the order form and order review pages look good since customers will see them.

Stage 3

Languages: HTML, PHP, and MySQL

First you will need to add an extra column on your order table in the database. That column will be defaulted to false and get marked true when the order is fulfilled.

Add an extra column onto the table that has a form and button to fulfill the order. You can call back to the same page that you are on currently with a get parameter for the id of the order that you want to fulfill.

<form method=”GET” action=”this_order_list_page.php?fulfillOrder=<order_id>”>

Stage 4

Languages: HTML, PHP, and MySQL

This step will require some creative database querying to get out all of the orders that you need over the specific time frames. You may also need to add a column to your order table to tell when the order was submitted if you didn’t do that before.

Make a page which shows the amount of money and number of orders you got over each of the last 7 days in one table, each of the last 7 weeks in another table, and each of the last 7 months in another table

Stage 5

Languages: HTML, PHP, and MySQL

If your toppings are in a field on your order then you may want to create two other tables: toppings and order_toppings. Toppings will have the name of each topping and the price. Order_toppings will have the id of the order and the id of the topping. It is a “join table” a table that says which toppings are on an order.

If you have it split out you can simply query the whole order_toppings table and “GROUP BY topping_id” and “COUNT” to get the number of each topping. See this blog post: http://www.w3resource.com/mysql/aggregate-functions-and-grouping/aggregate-functions-and-grouping-count-with-group-by.php