**Javascript**

Dynamic Recipe Project

A company is creating a website featuring a variety of recipes. They would like their recipes to have some dynamic functionality.

**HOME PAGE -  The following guidelines have been suggested.**

1. Display available recipes for the customer to review

* The page should display the available recipes
* The display format is up to you
* The customer can select the recipe they would like to see displayed

2. Display the three following main areas of the selected recipe. This includes suggested functionality.

* Only one recipe should be displayed at a time
* All the content for a recipe should be contained on the same page

The **recipe heading** should:

* Display the name of the recipe
* An image representing the recipe.
* The number of servings for the recipe
  + The customer should be able to change the number of servings
  + Options should include:
    - normal quantity - default
    - half quantity
    - double quantity
    - Additional options would be beneficial! Example: by servings, number of people, etc.
  + This will change the quantities in the ingredient list
  + Radio buttons were suggested, but more user friendly options would be better.
* The average preparation/cooking time for the recipe.
* The difficulty of the recipe.

The **recipe ingredients list** should:

* Should show when the customer clicks on something to show the list.
* Display a list of ingredients with quantity followed by the ingredient
* Quantities may need to change based upon Number of Servings.
* Be able to close the ingredient section.

The **recipe instructions list** should:

* Show when the customer clicks on something to show the list.
* Make sure any quantity changes, as needed are reflected in this area as well.
* Be able to close the instructions section.

**Input Page - A customer would use this page to input a new recipe into their application**

The form should consider the following things:

* Allow the customer to provide the heading information
* All the customer to include an appropriate image
* Allow for the input of an unknown number of ingredients
* Allow for the input of an unknown number of instructions
* The form will require data validation as needed
* When complete the form will create a Javascript Object in an defined format
* The new object will be stored in local storage.

**Data Storage**

* The input form will store its data into a predefined Javascript Object
* It will store the object using local storage
* The Home Page will access the objects from local storage to display details

**Technical Instructions**

* This a custom coding assignment.  The use of any CMS tools is not allowed.
* All pages must be responsive. You are welcome to use custom Media Queries along with Flexbox or Grid, Bootstrap or any other common CSS Responsive Framework.
* You are welcome to combine this project with another course project to expand its capabilities. A combined project must meet the project requirements for each course.
* Style and presentation are important for this project.

**PHP**

Your final project is about your PHP/MySQL skills.  It must be coded by you and implemented into your website by you. You cannot use a pre-built CMS tool of any type for this project such as Wordpress, Joomla, Drupal, etc.

Use one of the following methods as the foundation for your final project

1. A new site using the skills presented in the course.
   * The site and content can be of your own choice.
   * I recommend using one of your existing projects or layouts from other courses
2. Build the server side processor for the Recipe Manager project from WDV321 Advanced Javascript.
   * This project can choose to use server side formatted content. This works well if you hard coded your HTML content. You can replace it with content formatted in PHP. **OR**
   * The server processing can send JSON content to the client for consumption using AJAX calls. This assumes your Recipe Manager uses JSON objects for recipe content.
3. Use the Event Processing examples from our course assignments.
   * Format and bundle the homework assignments for Events into a comprehensive package that meets the project requirements listed below.
   * Grading for this project will be reduced compared to options 1 and 2. (265 points rather than 300, but extra credit is available to get the full 300 points)

**Project Requirements**

**The public facing portion of your website applications.  This should look and act like a website application.**

1. Your web application must contain a header and a footer.
   * The header can contain “WDV Final Project” or a more fun name based on your content. Whatever you’d like.
   * The footer should contain © 2021
     + Please make the year dynamic by getting the current year using PHP
2. Dynamic content page(s)
   * One or more pages of your website should pull product/event/content information from your database.
   * The content should be formatted by PHP according to your site presentation and displayed on the page.
3. Contact Page
   * Must have a simple contact form that accepts name and email. Also include a “Message” field that is an HTML textbox
   * When submitted, this form will send an email to you with the user’s information. Please also send a copy of this email to the person that submitted it (that way I can get a copy and make sure your form is functioning).
   * When the email is submitted and the page reloads, the form should be gone and the user should see a personalized Thank You message letting them know their form was submitted successfully
   * Form needs to be protected using the Honeypot method.
   * 5 points extra credit: Make all fields required
   * 10 points extra credit: If the honeypot is invalid, display an Access Denied message rather than a success message with a button that reloads the contact page of your web app.
4. A link to sign in to the application to access the Administration functions
   * This could be on your navigation bar.
   * Or look to website applications that you sign into for layouts and placement ideas.
   * This will link to a log-in page that requires username and password.

**Administration Side of your application. Only available to those with administrative rights to change website content**

1. Login Page
   * Should be styled similar to your application but with enough differences that you are in a separate part of the website.
   * Accepts username and password.
   * Validates against your database.
2. A landing page/Admin System page.
   * Display options to Add records to the database or View all records (described below)
   * Establish and use session variables to protect your pages from unauthorized access.
3. The following pages/functionality will be provided for the logged in administrator of the website application.
   * Add Page
     + This page will contain a simple form with all of the fields required to add a new record to your database
     + Your PHP will use the INSERT command to insert the new record into your database
     + The form will be protected using the Honeypot method
       - If the honeypot fails, simply reload the page and don't process the form. It'll be as if nothing happened
     + When an item is inserted, the page will refresh and show a success or error message
     + This form must use the self-posting method
   * View All Page
     + This page will use the SELECT command to get all records from your DB (database) and display them on the page. (you don’t need to use <table> tags if you don’t want to. You can use any HTML/CSS that you’d like as long as it’s nicely formatted on the page and easy to read)
     + Each item will have a delete button that will use the DELETE command to remove the item from the database
       - Upon deletion, the page will refresh and show a success or error message
     + 20 points extra credit: Give each record and update button that uses the UPDATE command to update the record
   * Update Page (If doing extra credit)
     + This page will show a pre-populated form with the values from the database for the current record
       - Use the SELECT command to select only the record that is being updated
       - I will leave it up to you to figure out how to pre-populate the fields in the form, but am available to guide you in the right direction if needed.
     + Upon update
       - If an Error occured, reload the page and display the error
       - If successful, take the user back to the View All page with a success message that the specific record was successfully updated
4. Technical requirements
   * All view pages must be responsive.  You are welcome to use any combination of responsive design frameworks or layouts such as Flexbox, CSS Grid, Bootstrap, Foundation, etc.
   * Insert, Delete and Update forms must use the self posting process.
   * Your PHP must implement OOP.
     + For example, if your web application displays video game data you could have a class called Video\_Game. And then use that class to CRUD (Create, Read, Update and Delete)
     + Standard PHP functions don't need to be put into classes and can be used as is.
   * All pages should implement MVC concepts for separating processing from presentation
   * All SQL queries must use PDO and prepared statements.
   * It is expected that your site reflect production quality.  Presentation of the site, including the admin area will affect your project grade.
   * User Interface and User Interactions are important. Consider this when working with format, layout and messages.
   * Your project must include production error handling using PHP log.
     + Your logs can go to the default log file that php error\_log points to, or a custom log file like we defined in class
   * Difficulty and complexity of your project will affect your project grade.
     + The more advanced you go, the better. But the more advanced you go, the more organized your code must be.
     + If you go too simple and do the bare-minimum, it can affect your grade negatively.
5. Your final project will be stored and available on your **Git Repo Account**.
   * Include your Git repo link when submitting this project on Blackboard.
   * Include a link to your WDV341 homework page when submitting this project on Blackboard.
     + By this point your homework page should include links to all assignments
   * Provide a default username and password of "wdv341' for testing purposes.