Healthy Eats Website Technical Design Documents 1/21/2017

Purpose of website and project goals

The purpose of this website is to allow an easier way for users to fine healthy recipes and to create a quick and easy shopping list that will allow them to make these specific recipes that they have chosen. Healthy Eats will allow someone to have a basic log in to keep their favorite recipes and to even add ones if they wish. Recipes will not be screened therefore if they are not healthy it will be on user. This is simply a site that will allow the sharing of recipes in an easy and covenant form. This is a great way for health enthusiast around the world to connect. Major supplement providers such as GNC, Vitamin shop and bodybuilding.com will be very interested in this program. The recipes will have a calorie count posted by the user who added the recipe. The basis is on trust however, if the recipe calls for whole mile and or real sugar and butter this is not a healthy recipe. The basics of the website needs to be as simple as possible allowing all levels of computer users to navigate through the options. This website is a free application on pc only not utilized for phone allowing the ease of access for new computer users. The backend will be a Dynamic Database Populator. Since this is being built for programmers to use. The focus will be on the code behind and the options available for adding different data as opposed to the user experience. Warnings and errors will be in color but overall the program will be a simple windows form.

Different types of Systems and instructor being used

The website will be created using html, css, JavaScript, php. The code will be organized into files listed below:

- Basic Files Utilizing Materialize
  - css
    - Utilizing martialize from web page development
  - img
    - File for all images and video content for web page's
  - Js
    - Materialize for JavaScript specifically materialize.js, materialize.min.js.
- Following are all html files
  - Index.html
    - This is a home page allowing user to login in and or register to web site and for ease of access from tabs to other web pages. This Utilizes buttons, tabs, and navigation bar including footer.

- Classes:
  - Nav-extended creates a navigation bar.
  - Page-footer creates an interactive footer.
  - Instructions: of website

- mylist.html
  - Includes tabs for ease of access for other pages showing a printable list. Includes navigation bar and footer.
    - Classes:
      - Nav-extended creates a navigation bar.
      - Page-footer creates an interactive footer.
      - List: of all ingreadiants

- recipies.html
  - Utilizes category search from dropdown box (Breakfast, Lunch, Dinner, Snacks). Showing recipes from cards, and navigation bar, footer included.
    - Classes:
      - Nav-extended creates a navigation bar.
      - Page-footer creates an interactive footer.
      - Recipe cards: cards with recipies on it with interactive check box.
- login.html
  - Basic web box with user_id, user_email log in.
    - Classes
      - Userid:
      - Useremail:
- login.php
  - Php connection to mysql database for user. Using user_id and user_email.
- register.html
  - Registration for user creating user_id, and user_email
    - Classes:
      - userid_registration: users id
      - useremail_registration: user email
      - userid
      - useremail
- register.php
  - Php connection to mysql creating user_id, and user_email
- addrecipie.html
  - Allows user to add recipes buy using basic input form
    - Classes:

        - Ingredient1
        - Ingredient2
        - Ingredient3
        - Ingredient4
        - Ingredient5
        - Ingredient6
        - Ingredient7
        - Ingredient8
        - Instructions

    - utilities.php
        - Connection to mysql database using php for mylist.html, recipies.html, addrecipie.html.

Website Structure

The basic structure of the website will have three pages attached to it. The three front end web pages will consist of:

One the home page showing an overall view of what the website consists of and how to use it through an easy step by step guide. The guide will describe base use of this website and reasoning behind it. The home page will have a basic login pop up window allowing the user to login and or register making an easy profile. Because this is not a heavily watched free site the level of security will be very low. No personal information will ever be asked or utilized on this. The login will simply ask for a user name (With a cap of ten) and a password (No special characteristic needed). On the navigation bar the user will be allowed to tweet and or Facebook the following website. The basic navigation bar will have three tabs Home, My List, Recipes.

The second page “Recipes” will have a list of recipes. There is a simple drop down sort box allowing the user to choose from the following meal times: Breakfast, Early Morning Snack, Lunch, Afternoon Snack, Dinner, Late Night Snack. The user that is logged in selects recipes that they wish to make then check-out. The checkout button will take them directly to the last page. These recipes will be down loaded from users allowing a cross interaction with users.

The last Page will consist of a checkout and or grocery list page. It will simply show the users their selected recipes need in the form of a grocery list.

Database Populator

The database populator will be created using C# with ADO.NET. The code will be organized into files like so:

- DBPopulator.cs – main class all initial classes and methods are called through here. The code to move through the program (buttons, text boxes, etc) will be controlled here.

- DBConnection.cs – database connection class. This class will contain the SQL connection string to connect to a user's database. The class will also create the Dataset

- DSSchema.cs – *inherited from DBconnection.cs* this class will handle getting all of the information associated with the databases, tables, and columns in the dataset. These will be stored in lists that will be called at different points in the program

- DSInsert.cs - *inherited from CSScema.cs* this class will insert the data into the appropriate database.

- Validator.cs – standard validation class.

Since the program uses C# with ADO.NET naming conventions will adhere to the industry standards. UpperCamelCase will be used for classes and methods while lowerCamelCase will be used for all variables and functions. The exception to these standards will be the use of Hungarian notation for naming Windows Form Controls. Microsoft recommends writing out the entire control description (ex. buttonLogin); instead the description will be shortened to those commonly used in the industry (ex. btnLogin).

Additional Information

http://ict.neit.edu/001393899/CapstoneWebsite/index.html