**CS 3320: Internet Software Development**

**Project and Assignments Description**

**Description:**

You have been asked to build a web application for e-commerce shopping cart.

**Here are additional details:**

1. Website contains four sections:
   * User Information
   * Shopping Cart
   * Shipping Information
   * Checkout with Payment Information
2. There are the fields for each section:

|  |  |  |  |
| --- | --- | --- | --- |
| **Section** | **Field** | **Required** | **Notes / Additional requirement**  **(We will be adding some of these iteratively in assignments and project)** |
| User Information | Full Name | Yes | Maximum 100 characters |
| User Information | Address 1 | Yes | Maximum 100 characters |
| User Information | Address 2 | No | Maximum 100 characters |
| User Information | City | Yes | Maximum 100 characters |
| User Information | State | Yes | * Drop Down list. * User selects one state. * Store State Codes in DB and use AJAX to load the drop down when page loads. |
| User Information | Zip Code | Yes | Maximum 9 characters, minimum 5 characters |
| User Information | Phone | Yes | Maximum 12. Use RegEx to validate this field.  Format: ###-###-####  Show format tooltip next to this field. |
| User Information | Email | Yes | Maximum 100 characters.  Use RegEx to validate email address.  Provide proper alert message if format is incorrect:  e.g. “Please enter a valid email address.” |
|  |  |  |  |
| Shopping Cart | Product | Yes | * Drop Down list. * User can select multiple products. * Store product data in DB and use AJAX to load the drop down when page loads. * Drop down should show list in this format: Product (Unit Price)   e.g. Banana ($ .50 / lb) |
| Shopping Cart | Units | Yes | User enters how many units.  e.g. 4 |
| Shopping Cart | Unit Price | Auto Populated | Unit price is auto populated once user selects the Product |
| Shopping Cart | Total Price | Auto Calculated | Once user enters units of Product JavaScript calculates the Total Price and populates this field. |
| NOTE: Add shopping cart items in a HTML table after user selects to add item. | | | |
|  |  |  |  |
| Shipping Information | Use Physical Address | Checkbox | If user checks this option. All the shipping fields auto populated with the user address data from user information. |
| Shipping Information | Address 1 | Yes | Maximum 100 characters |
| Shipping Information | Address 2 | No | Maximum 100 characters |
| Shipping Information | City | Yes | Maximum 100 characters |
| Shipping Information | State | Yes | * Drop Down list. * User selects one state. * Store State Codes in DB and use AJAX to load the drop down when page loads. |
| Shipping Information | Zip Code | Yes | Maximum 9 characters, minimum 5 characters |
|  |  |  |  |
| Checkout | Total Shopping Amount | Auto populated | JavaScript / jQuery automatically calculates total amount by adding shopping cart items. |
| Checkout | Total Tax | Auto populated | JavaScript / jQuery automatically calculates total shipping charges due = total shipping amount \* .08 |
| Checkout | Total Shipping Charges | Auto populated | JavaScript / jQuery automatically calculates total shipping charges due = total shipping amount \* .03 |
| Checkout | Total Amount  Due | Auto populated | JavaScript / jQuery automatically calculates total amount due by adding shopping amount + tax + shipping charges. |
| Checkout | Payment | Dropdown | VISA, MasterCard, American Express |
| Checkout | Card Number | Yes | Maximum 16 characters. JavaScript validates only numbers are allowed and 16 characters are entered.  You can use RegEx. There are plenty examples. |
| Checkout | Card Expiration | Yes | JavaScript / jQuery validates the format MM/YYYY |
| Checkout | Agreement | Checkbox and required. | Verbiage: “I agree and authorize the amount to be charged to my credit card.” |
| Checkout | Submit Order | Button | When user clicks the Submit Order button, you submit the order by using JavaScript / jQuery to create a confirmation page that shows the order summary. |
|  |  |  |  |

**Deliverables:**

* Install and configure XAMPP. You will need this for PHP, database, and web server.
  + <https://www.apachefriends.org/index.html>
  + Watch this video to understand how XAMPP works: <https://www.youtube.com/watch?v=5xnXHLHR3AE>

1. **Assignment 1 (Due: 09/20/2019)**
   1. Create html pages for each section described above.
   2. We are only building html pages in this assignment.
   3. You can use GIT to collaborate. University provides free git accounts to students. You are free to use open source version control tools as well like github. Or you can use google drive and provide a link so we can view your code.
   4. Provide screenshots of each page you developed in a word /pdf document.
   5. HTML files.
   6. Grader should be able to view your code and should be able to view it in browser.

1. **Assignment 2 (Due: 10/04/2019)**
   1. Put all html pages together by creating a html layout page for your website.
   2. Template example: <https://www.w3schools.com/html/html_layout.asp>
      1. It’s only an example. You are not restricted to use just this layout.
      2. Layout should be user friendly and appealing.
      3. Use CSS styling to beautify your website.
      4. All CSS code should be in a file. Do not add it inline or in header. Add a link to CSS in HEAD. Examples are available in class notes or google it.
      5. Layout must contain header, footer, navigation menu, and content area.
      6. Only content area should change when user clicks on a menu link.
      7. User should be able identify which link they are viewing.
   3. Provide screenshots of each page you developed in a word /pdf document.
   4. HTML files, CSS file.
   5. Grader should be able to view your code and should be able to view it in browser.
2. **Assignment 3 (Due: 10/18/2019)**
   1. Add JavaScript to your website.
   2. Use JavaScript/jQuery to complete the requirements stated above.
   3. Provide screenshots of each page you developed in a word /pdf document.
   4. HTML files, CSS file, JS / jQuery file.
      1. All JS code should be in a file. Do not add it inline or in header. Add a link to JS in HEAD. Examples are available in class notes or google it.
   5. Grader should be able to view your code and should be able to view it in browser.
3. **Assignment 4 (Due: 11/01/2019)**
   1. Database to store and retrieve
      1. Database name: cs3320
      2. Use above data on page 1 & 2 to decide the column type and length.
      3. Table: UserCredentials (userId, username (50 chars), pass (50 chars))
      4. Table: UserInformation (userId, fullname, address1, address2, city, state, zip)
      5. Table: ShippingInformation (userId, address1, address2, city, state, zip)
      6. Table: PaymentInformation (userId, cardType, cardNumber, expDate,)
      7. Table: Products (productId, description, unitPrice)
      8. Table: Orders (userId, orderNumber, productId, quantity, totalPrice)

userId, orderNumber, productId is unique combination

* + 1. Table: States (SQL here: <https://gist.github.com/JeremyMorgan/5833666> )
  1. Add AJAX to your website.
  2. Use AJAX to complete the requirements stated above.
  3. Provide screenshots of each page you developed in a word /pdf document.
  4. Provide screenshot of tables you created.
  5. HTML files, CSS file, JS / jQuery file, php files.
     1. All JS / AJAX code should be in a file. Do not add it inline or in header. Add a link to JS in HEAD. Examples are available in class notes or google it.
     2. NOTE: for AJAX to work properly you will need at least two DB tables, i.e. Products and Lookup. Create sample product data. Example is provided in the video link I provided above.
     3. Example here: <https://www.w3schools.com/php/php_ajax_database.asp>
  6. Grader should be able to view your code and should be able to view it in browser.

1. **Project (Due: 12/03/2019):**
   1. Create User Login Page. Allow user to enter username and password.
      1. Create some usernames and passwords in database for testing.
      2. Authenticate user against database. If user enters correct credentials, then take user to landing page that you created in assignment 2.
      3. If user didn’t enter correct credentials show proper message “Username and Password combination incorrect. Try again.”
      4. Store userId in a session object. You will need userId later when user submits an order. Example here: <https://www.formget.com/login-form-in-php/>
   2. Add a Link in Menu “Create a New Order”.
   3. Create New Order opens User Information Page.
      1. If user created any orders in the past, then retrieve User Information from database using php and auto fill User Information. User can change this data if they want.
      2. Provide a button “Next” on User Information Page.
      3. When user Clicks on Next validate all required fields are filled using JavaScript or HTML5 validation. If all fields are complete, then only take user to Shopping Cart Page. If any data is missing show proper message and stay on User Information page.
      4. User add / removes items on Shopping Cart and when ready to checkout, clicks on “Checkout” button provided at the bottom of Shopping Cart page.
      5. Now user comes to Shipping Information Page. User enters shipping Information and Proceeds to Payment page by clicking Next.
      6. On Payment Information Page user provides payment Information.
      7. Before submitting the order, user can go back to any page to make changes. Provide Previous button at the bottom of each page to allow this feature. User Information Page will not have Previous button since that’s the first page in Order.
      8. To complete the Order, user clicks on “Submit Order” button provided on Payment Page. Use php to submit order to database.
      9. When user Clicks on “Submit Order” data is inserted to database in proper tables.
      10. After successful data insert User sees the Order Summary.
   4. At this point User can Create Another Order or Exit.
   5. When user Exits, user session needs to be cleaned. Example is here: <https://www.formget.com/login-form-in-php/>

* Demo
  + Each team is required to do a demo.
  + Demo schedule will be provided later.

What to turn in:

* Working code, html files, css file, js file, php files, screenshots of each page showing an Order from start to summary page.

**Suggestions:**

* Here are some shopping cart examples:
  + <https://phppot.com/php/simple-php-shopping-cart/>
  + <https://speckyboy.com/free-shopping-cart-css-javascript/>
* Ask for clarifications if something is not clear.
* Seek continuous feedback.
* If you make any assumptions provide reasoning behind it.
* If you feel some information can be added to any of the sections to make it user friendly, please do so.