1.1 User's Manual

1.1.1 Basic User/Testing Employee

Getting Started
By default, our webpage begins at http://40.71.228.49/StorefrontTestingApp/login.php, however; depending on deployment by a local system administrator, the target address will be different and it will be the responsibility of your system administrator to provide you the correct address. In further explanations, this IP address prefix will be used, so in the event of using a different origin address, adapt accordingly. Functionality descriptions are based on the bug-free deployment of the webpage, as such, when bugs are assigned, it is possible that some descriptions are intentionally no longer valid. However, core functionality of both registration and login are exempt from bugs as this would prevent the ability to enforce that users must be logged in to use the system or that a bug could be assigned to a specific logged in user. Some additional functional explanation is provided in the following sections in order to make clear the intended behavior to help differentiate between what may be an intentional bug assigned to you and what may have been an oversight on our behalf.

Registration
In order to access the contents of the website you are required to be logged in, as such, if you do not currently possess an account of your own or are not provided with login credentials you will need to navigate to http://40.71.228.49/StorefrontTestingApp/register.php to create an account. You will be required to enter a valid email address and submit your desired password. Once filled out, you will click "Sign me up!". This will send a verification email to the email address you provided containing a link that will complete the registration process. After completing this step you will be redirected to a page where you will be prompted to provide basic personal information such as name and address. Upon completion would will be taken to the storefront homepage.

Login
Assuming that you already have an account that has been completed in line with the previous registration steps, you can navigate to http://40.71.228.49/StorefrontTestingApp/login.php and type in your email and password into the appropriate fields. Upon successful login you will be taken to the storefront homepage.

Storefront Homepage
The main purpose of the homepage is to show a list of all of the products along with a number of filtering options to help you to find the target item. The page contains three major filters, first is the search box present at the top of the page in the header that is present on all pages after login. Search input can be submitted either by pressing the enter key while the search box is the active element or by clicking the search icon in the search box. Search will show only products whose names are similar to your search query. Price filtering is handled by two slider bars, min price and max price, located in the filters section of the sidebar. Both slider bars have an attached input text box that both displays the value associated with the slider bar position as well as allows you to input to specify exact values. Upon releasing the mouse on the slider bar, pressing enter in the input box or incrementing the value in the input box, the selection of products will be filtered accordingly. Finally is a category selection filter, also located in the sidebar, which gives you the ability to display only the category of products which you would like to view. All of the filters can be used together. The final selection options is a sort by dropdown box which controls how the products displayed are sorted. The products shown each contain an

image, name, price, description and a configure button which will redirect you to the product page which contains all variations of that item that are present in the product database.

Page Header
Each page after login has a header bar that spans the top of the page. This bar contains a home button which will take you back to the product list page, a search bar which will take you to the product list page and return the result of the search filter, an account button, an orders button and a cart button. The account button expands when moused over revealing a logout button and an account page button which will redirect you to the account page. The orders button will take you to a page that will allow you to look at past completed orders and the cart button will take you to your cart page.

Product Page
The product page displays the product image, name, description, price, and stock as well as the selection buttons for color, size and sku of the currently selected item. In the event that there are multiple color, size or SKU variants for the same item, additional selection options will be shown for color, size or SKU. Clicking these buttons will redirect you the product page associated with the selection. Finally there is an add to cart button along with a quantity selection input box. Clicking on the add to cart button will that however many items to the cart in line with the value in the quantity select box. Stock quantity is considered when adding to the cart so in the event that the number of items added to the cart combined with the number of that item currently in the cart exceeds the stock, the add to cart request will be denied. Basic feedback is provided when clicking the add to cart button but to view the cart contents, you must use the cart button from the header to view details.

Cart
The cart page details all of the products that are currently in the cart. All products have a quantity selection option in the event that you wish to alter the number of an item in your cart, however once changes are made, you must click the “Save” button or the “Checkout” button located at the bottom right of the cart contents listing. To remove an item completely from the cart, changing this quantity to zero and clicking save will remove the item from the cart. Below the list of cart contents is the total price of the cart including tax. Once you are satisfied with the contents of the cart and wish to checkout, immediately beside the save button is a checkout button which will redirect you to the checkout page where order information is finalized.

Checkout
The checkout page contains input forms for billing information as well as shipping speed selection. Changing the shipping speed from the standard rate will incur an additional charge and the total order price is displayed. Once all information has been provided, clicking the checkout button will place the order and take you to a receipt page which highlights all of the details of the order.

Account Page
This page can be gotten to by way of the header, mousing over the Hello, <Your Name> button will reveal a button called view account. This page contains a listing of all of the account information you have provided. In the event that any of the information needs to be changed, by simply changing the field of interest and clicking the update button at the bottom, the newly input data will be updated in the system.

Past Orders
Also linked in the header, this page can be accessed by clicking the order button in the header. This page will display a minimalistic display of all of your orders, each with a button to view

additional information related to this order. Clicking on this button will display a full list of the order contents as well as the price of the order.

General
All interactive elements such as buttons, input fields and slider bars possess an element id unique to the page so that automated testing can be done to interact with specific and consistent page elements. While assigned bugs will vary, the most common type of bug will be that interactive page elements do not respond as anticipated or described in the sections above.

1.1.2 Storefront Administrator

Bug Assignment
The bug assignment page displays all of the registered users and all of the bugs in the database. By clicking into the user list, the list of users will be displayed as “Last Name, First Name”. You can type into the user list, and it will automatically display matching entries as you type. The list of bugs is organized alphabetically by the functional area, or page and feature, that the bug affects. You can multiselect by holding the control button while making your selections. Bug assignments can have start and end dates. By default the end date is null, but the start date will default to the day you are assigning the bug.

When you click Review Bug Assignment, the user and the bugs selected during this session will be displayed. Clicking the Save Assignment button will add these bug assignments to the database. If the bug assignment already exists for that user, you will receive an alert as it continues to save any remaining bugs.

Updating Bug Assignment
The update bug assignment page displays a table of bug assignments when a user is selected. The table will allow you to change the start and end dates for an assignment and then either update the assignment or delete the assignment from the database.

Bug Review, Editing and Creation
This page allows you to view all of the existing bugs along with their details and relevant code block as well as allowing the insertion of new bug definitions. By clicking on the field on the left side of the screen that says “Enter a bug name” you will see all of the bugs that are in the system. This will populate the fields on the right side of the screen with the details of name, functional area, descriptions and code block. In the event that any of this information needs to be edited, simply changing the contents of these fields and clicking the “Update Bug” button will update the information saved. In the event you wish to add a new bug, typing in values for bug name, functional area, description and code block with no bug currently selected and pressing the “Save New Bug” button will save this bug into the system. However, due to the nature of hooking the bug into the code, the insertion of a bug will still require making a few simple additions to the core php files in order to support the bug. This process is described in detail in the System Administrator’s Manual. Therefore, **if you are not also a system administrator or are not working directly with a system administrator during the process of adding a new bug, you should not use the “Save New Bug” feature.**

1.2 System Administrator's Manual

Getting Started
Our project was developed for use on a LAMP stack (Linux, Apache, MySQL, PHP). Assuming that this requirement has been met, simply navigate to the directory that you wish to use and

from the terminal type “sudo git clone https://github.com/nawa236/StorefrontTestingApp.git”. This will download all of the required files for execution and you may continue to the next step.

Database Initialization
This project relies on the existence of a mysql database accessible by the server where the code is executing. In order to function, the database connection attribute will need to be updated as appropriate for your environment. Both in the file “dbConnect.php” and “database.php” are variable definitions for servername, username, password and database name. These variable will need to be manually changed to match the appropriate login credential for your mysql. Once the values have been altered, navigating to http://40.71.228.49/StorefrontTestingApp/index.php substituting the initial ip address prefix for the location of your server and directory path. Assuming that the message “Welcome to the Employee Store.” is displayed, the database settings have been properly provided and the initialization of the database for use is complete.

Account Verification Email host
The PHP server the website is hosted on must have a mail sending or relay service that is properly configured for the website verification mailing options to function correctly. This will be highly variable depending on the current server configuration and desired functionality, so it will be left to the administrator to determine the best path to take regarding this service.

In addition, the mailing system was tested by the team using a relay to the gmail smtp servers with the account trissentialbugsite@gmail.com. There are references to this email (as the from entry for the email) in register.php, forgotpassword.php and reverify.php. These entries will need to be changed depending on the intended email to be used for this purpose.

Additionally, the email link is currently setup to use our webserver but will need to be changed to the server IP of the PHP server the website in installed on.

If it is desired to use the gmail account in the future, the details are below:

Account: trissentialbugsite@gmail.com

PW: BugsAreFeaturesToo

Maintenance - Adding New Bugs
The primary functionality additions for this project will be the ability to add new bugs into the system. As mentioned in the Storefront Administrator section of the user manual, adding a new bug is composed of two major tasks. The first should be to determine what you want the bug to do and to navigate to the appropriate php file that contains the page element that you wish to add a bug to. Once on the page, select the line or lines of code that you wish to add a bug for and encapsulate it similar to this example. *Text in “< >” used to describe the contents.*

```
//***** Bug Start *****//
$bugCode = bug_check(<bug id>);
if(is_null($bugCode))
      <Original Line/Lines of code>
else
      eval($bugCode);
//***** Bug End *****//
```

The example above is for adding bugs to a php section of code and should not produce an error as long as the php includes either header.php or bugCheck.php. If neither file is included, you

will need to include bugCheck.php. While similar, because bugCheck function is a php function, in the event you wish to add a bug to a JavaScript section of code, an example would look like:

```
//*****  Bug Start ****//
var bugCode = "<?php echo bug_check(<bug id>);?>";
if(bugCode == "")
      <Original Line/Lines of code>
Else
      eval(bugCode);
//*****   Bug End  ****//
```

In many cases you may not need a branch for both bug or no bug, as you may want to add a line of code if there is a bug, or only run the correct line in the event of no bug, without explicitly having additional bugged code to run. All of these methods are acceptable, so consider what is needed when actually inserting the bug branch handling code.

The other task is to add this bug into the database. This can be accomplished through the web interface by using the admin.php page; however, as this is just a GUI for interacting with the database, it is also possible to manually insert new rows into the bug table in the database. On the admin.php page clicking on the “Bug Creation and Editing” tab will allow you to enter in a new bug name, functional area, description and code block. Clicking the “Save New Bug” button will add this bug definition to the database. Once saved in the database, it is now possible to go back to the “Bug Assignment” tab and select users to assign this bug to.

Maintenance - Adding New Web Page Features

All existing web pages are designed as single php files that when needed, call auxiliary php files through ajax or are “connected” through the use of page navigation hyperlinks. Due to the lack of inherent coupling, editing or adding features to these pages should be easy. While exact method is up to you, simply adding the additional elements into the HTML section of the php file, ensuring that the output is as expected, then adding whatever dynamic page interaction is needed is likely the best course of action. In the event of making new pages, unless there is a need for your new pages to directly interact with a previous page, such as through get/post, it is recommended to update or query from the shared database for most inter-page interactions.