Group name \_\_\_WEEBSYSTEMS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Members: \_\_\_Jareed Banta, Piolo Pagdagdagan, Nicole Mae Leyva, Maxene Mostoles\_\_\_\_

Course Yr. & Block: \_\_BSIT ANIMATION 3\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| Problem Number: 1 |
| Problem Statement |
| Need to make the display of items coming from the database |

|  |  |  |
| --- | --- | --- |
| CODE CHANGES | INTERFACE CHANGES | Describe the functionality added or changed |
| The code now displays items fetched from the database, which were not showcased during our defense because items were previously hard-coded instead of being inserted into the database.  The system can now fetch items from the database using SQL queries and display them on the user-facing shop page.  When logged in as an admin, the Admin Dashboard provides options to insert and delete items.  An admin panel has been added with full CRUD functionality for managing products. | The interface has been updated to separate admin and user accounts.  **Admin Account:**  When logged in as an administrator, you will now have access to the Admin Dashboard, which includes the following features:   * **View All Products:** A comprehensive list of all products with options to edit or delete them. * **Add New Products**: An intuitive form for quickly adding new items to the database. * **Manage Inventory:** Track product stock levels and update details as needed.   **User Account:**  Regular user accounts now have a simplified interface tailored for shopping. Users can:   * Browse through the list of available items. * Add desired products to their cart. * Proceed to checkout for purchasing. | **Admin Account:**  The Admin Dashboard now includes full CRUD (Create, Read, Update, Delete) functionality. Administrators can:   * Add, edit, or delete items in the database directly through an easy-to-use interface. * View and manage all stored items efficiently.   **User Account:**  When signing in as a user, the shop page now dynamically showcases all items stored in the database. This is a significant improvement over the previous version, where items were hard-coded into the system and lacked CRUD functionality.  By storing and managing items in the database, the system ensures:   * Real-time Updates: Any changes made by the admin are immediately reflected on the shop page. * Scalability: Easily add new products or modify existing ones without altering the code. * Improved User Experience: Users can seamlessly browse and purchase items from an up-to-date inventory. |

|  |
| --- |
| Problem Number: 2 |
| Problem Statement |
| Need to make the add to cart function interact with the database |

|  |  |  |
| --- | --- | --- |
| CODE CHANGES | INTERFACE CHANGES | Describe the functionality added or changed |
| The **Add to Cart** functionality is now database-driven, resolving a limitation not showcased during our defense.   * **Database Integration**: The system uses a cart table to store product IDs, quantities, and user IDs. * **Improved Functionality**:   + If an item already exists in the cart, its quantity is updated; otherwise, a new record is created.   + The **View Cart** fetches data from the cart and products tables, showing product details, prices, and totals. * **Real-Time Updates**: Quantity adjustments modify the database, and item removals delete the corresponding records. * **Data Persistence**: Cart data is preserved across sessions and devices for a seamless user experience. | **Add to Cart**:   * A new button is available on the menu page, allowing users to add items to their cart. * Clicking the button triggers the add\_to\_cart logic, which stores the item in the database.   **Cart Page**:   * Displays items fetched from the cart table, including:   + Product Name, Price, Image, Quantity, Total, and Actions (Update/Remove). * The **Grand Total** is automatically computed and displayed for all items in the cart. | Add to cart is newly added with the following functions:   * Can add items from the menu that you selected to the cart. * Immediately directs you to the list of selected items on hold. * Views order summary of following items. Product, price, quantity, total. * Computes & displays grand total of multiple orders. * Can remove items & compute the total price of multiple orders of the same product. |

|  |
| --- |
| Problem Number: 3 |
| Problem Statement |
| The Checkout function |

|  |  |  |
| --- | --- | --- |
| CODE CHANGES | INTERFACE CHANGES | Describe the functionality added or changed |
| The new checkout function fetches data from the database, allowing users to place orders by providing billing, shipping, and payment details. The cart data is processed, and the order is stored in the database, clearing the cart after checkout. | The checkout page now includes:   * Pre-filled billing and shipping details. * A user-friendly form for payment selection. * A smooth checkout process that displays cart items, total price, and previous orders. | Checkout function added alongside the add to cart function. The functionalities are the following   * Automatically fills in billing, shipping, & payment details. * Smooth straightforward checkout process. |

|  |
| --- |
| Problem Number: 4 |
| Problem Statement |
| Admin – order management, item management, reporting sales |

|  |  |  |
| --- | --- | --- |
| CODE CHANGES | INTERFACE CHANGES | Describe the functionality added or changed |
| The code for this section of the system is new for the group has not submitted this part on the defense.  The code utilizes different fetch data function for each functionality.  fetch-orders.php for manage orders.  fetch-users.php for manage users.  And,  fetch-sales-data for view reports.  The data presented in the admin is connected to the database using db\_connect function. | The interface is similar to the user index, namely the navigation bar with the changes of the interface to suit the admin’s needs.  Both manage orders and manage users have the same interface. Only the view reports is different for adding a table chart to display the reports for total orders and income accumulated by the website. | Admin management newly added.  Functionalities included are:   * Views admin dashboard with the following: * Modify inventory by adding and deleting items from the shop. * Manage Orders- views pending & received customer orders. * View reports- checks website status & sales reports * Manage Users- views users information |