Description of improvements made:

Admin Panel

1. First of all I ensured successful connection to the database by specifically setting port number to 3307 because on my laptop I do not use the default 3306 port as it is used by another process.
2. I registered admin manually in the database and when I login as admin, trying to add a product produced the error below:

   ```
   Warning: move_uploaded_file(uploads/1732523384_plus-office-paper-
   staplerstaplersplus-japanscooboo-st-010eh4977564191269-213347.png): Failed to open
   stream: No such file or directory in C:\xampp\htdocs\purple-
   website\admin_dashboard.php on line 24
   ```

   ```
   Warning: move_uploaded_file(): Unable to move "C:\xampp\tmp\php45F6.tmp" to
   "uploads/1732523384_plus-office-paper-staplerstaplersplus-japanscooboo-st-
   010eh4977564191269-213347.png" in C:\xampp\htdocs\purple-
   website\admin_dashboard.php on line 24
   ```

   The above error simply means the **uploads/** directory could not be found so I added a PHP code to check if the directory exists and create it if necessary. I was able to achieve that using lines of code below:

   ```
   // Check if the directory exists, if not create it
       if (!file_exists($targetDir)) {
           if (mkdir($targetDir, 0777, true)) {
               echo "Directory created successfully.";
           } else {
               echo "Failed to create directory.";
           }
       }
   ```

3. After adding a product I added the following lines of code to refresh the page so that it shows an updated list of products. I did the same thing for deleting product.

   ```
   // Redirect to the same page to refresh
   header("Location: " . $_SERVER['PHP_SELF']);
   ```

```
exit;
```

4. **Deleting the product image when deleting product**: I added lines of code to delete a product image in **uploads/** directory when the product is deleted. The following is a complete code that handles product deletion.

```
// Check for form submission for deleting a product
if (isset($_POST['delete_product'])) {
    $productId = $_POST['product_id'];

    // Fetch the product details before deleting the record
    $stmt = $conn->prepare("SELECT image FROM products WHERE id = ?");
    $stmt->bind_param("i", $productId);
    $stmt->execute();
    $stmt->bind_result($imageName);
    $stmt->fetch();
    $stmt->close();

    // Delete the product from the database
    $stmt = $conn->prepare("DELETE FROM products WHERE id = ?");
    $stmt->bind_param("i", $productId);
    $stmt->execute();
    $stmt->close();

    // Delete the image file from the 'uploads/' directory
    $imagePath = 'uploads/' . $imageName;
    if (file_exists($imagePath)) {
        unlink($imagePath); // Delete the file
    }

    // Redirect to the same page to refresh
    header("Location: " . $_SERVER['PHP_SELF']);
    exit;
}
```

Customer Buying Process

1. Added dynamic products in categories.php: In the website, the user clicks “Shop” in the navbar to start buying process. Categories php file displays the products that a user can shop. Initially it was displaying static products, I improved the project by displaying dynamic products fetched from the database: The code to achieve the functionality is shown below:

```php
<?php
// Database connection
$conn = new mysqli('localhost', 'root', '', 'login_register', 3307);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Fetch products from the database
$sql = "SELECT * FROM products";
$result = $conn->query($sql);
?>
```

Code above connects to database and fetches products from the database.

```php
<?php
 // Check if there are any products in the database
  if ($result->num_rows > 0) {
  // Loop through each product and display it
   while ($product = $result->fetch_assoc()) {
     $productId = $product['id'];
     $productName = $product['name'];
     $productPrice = $product['price'];
     $productImage = $product['image'];
      ?>
     <!-- Product Item -->
     <div class="product-item">
      <div class="product product_filter">
       <div class="product_image">
         <img src="uploads/<?php echo $productImage; ?>" alt="">
       </div>
        <div class="favorite"></div>
         <div class="product_info">
          <h6 class="product_name"><a href="single.php"><?php echo $productName; ?></a></h6>
             <div class="product_price">$<?php echo $productPrice; ?></div>
      </div>
     </div>
    <div class="red_button add_to_cart_button">
<a href="#" onclick="addToCart('<?php echo $productId; ?>', '<?php echo
$productName; ?>', <?php echo $productPrice; ?>)">
Add to cart</a></div>
 </div>
<?php
}
} else {
 echo "<p>No products found.</p>";
}
 ?>
```

Additionally, I changed the “categories_custom.js” file to accept product id, name, image and price because product id is used to uniquely identify a product.

2. Created Orders, OrderItems, Addresses tables to store data related to orders appropriately.

```
CREATE TABLE `addresses` (
  `id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `address_line1` varchar(255) NOT NULL,
  `address_line2` varchar(255),
  `city` varchar(100) NOT NULL,
  `state` varchar(100) NOT NULL,
  `postal_code` varchar(20) NOT NULL,
  `country` varchar(100) NOT NULL,
  PRIMARY KEY (`id`),
  FOREIGN KEY (`user_id`) REFERENCES `users`(`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Above query was used to create addresses table.

```
CREATE TABLE `orders` (
  `id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `order_date` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `total_amount` decimal(10,2) NOT NULL,
  `status` varchar(50) NOT NULL DEFAULT 'pending',  -- Can be 'pending', 'completed', 'cancelled'
  `shipping_address_id` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  FOREIGN KEY (`user_id`) REFERENCES `users`(`id`),
  FOREIGN KEY (`shipping_address_id`) REFERENCES `addresses`(`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Above query was used to create orders table.

```
CREATE TABLE `order_items` (
  `id` int(11) NOT NULL,
  `order_id` int(11) NOT NULL,
  `product_id` int(11) NOT NULL,
  `quantity` int(11) NOT NULL,
  `price` decimal(10,2) NOT NULL,
  PRIMARY KEY (`id`),
  FOREIGN KEY (`order_id`) REFERENCES `orders`(`id`),
  FOREIGN KEY (`product_id`) REFERENCES `products`(`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Above query was used to create order items table.