COLLEGE OF ENGINERING AND COMPUTER STUDIES

**OUTCOMES EVALUATION # 2**

IPTL Integrative Programming and Technologies Lab

Submitted By

Stephen Grey E. Maano

Yves Isiah D. Ribao

Course & Section

BSIT 3-1

Date

January 16, 2025

**OUTCOMES OUTLINE**

1. **DESCRIPTION**

**This activity involves developing a functional e-commerce product management interface using HTML, CSS, and JavaScript. The task focuses on creating a dynamic and interactive user interface that simulates an online product management system commonly seen in e-commerce platforms. Implement features that facilitate the upload, display, editing, deletion, and organization of product data in a client-side web application**

1. **OBJECTIVES**

**• Create interactive web applications using HTML, CSS, and JavaScript. • Validate user inputs, including image file types and sizes, to ensure proper product uploads.**

**• Implement dynamic features such as editing and deleting products using JavaScript.**

**• Design user interfaces that focus on usability, responsiveness, and interactivity.**

**• Manage a product list by integrating features such as sorting, searching, and filtering.**

**• Integrate a product rating system that allows users to rate products from 1 to 5 stars.**

**• Demonstrate the ability to handle user actions and provide immediate visual feedback on the interface**

1. **THEORETICAL FRAMEWORK**

|  |  |  |
| --- | --- | --- |
| **INPUT** | **PROCESS** | **OUPUT** |
| * **User Input Product Information** * **Insert Images** | * **Save the inputted Information and the Image** | * **Display the Product Information and the Ratings** |

1. **RESULTS AND DISCUSSION** (screenshots of I/O, essential codes, logic with brief discussions)

Before

**A screenshot of a product upload box

Description automatically generated**

***Figure 1. HTML***

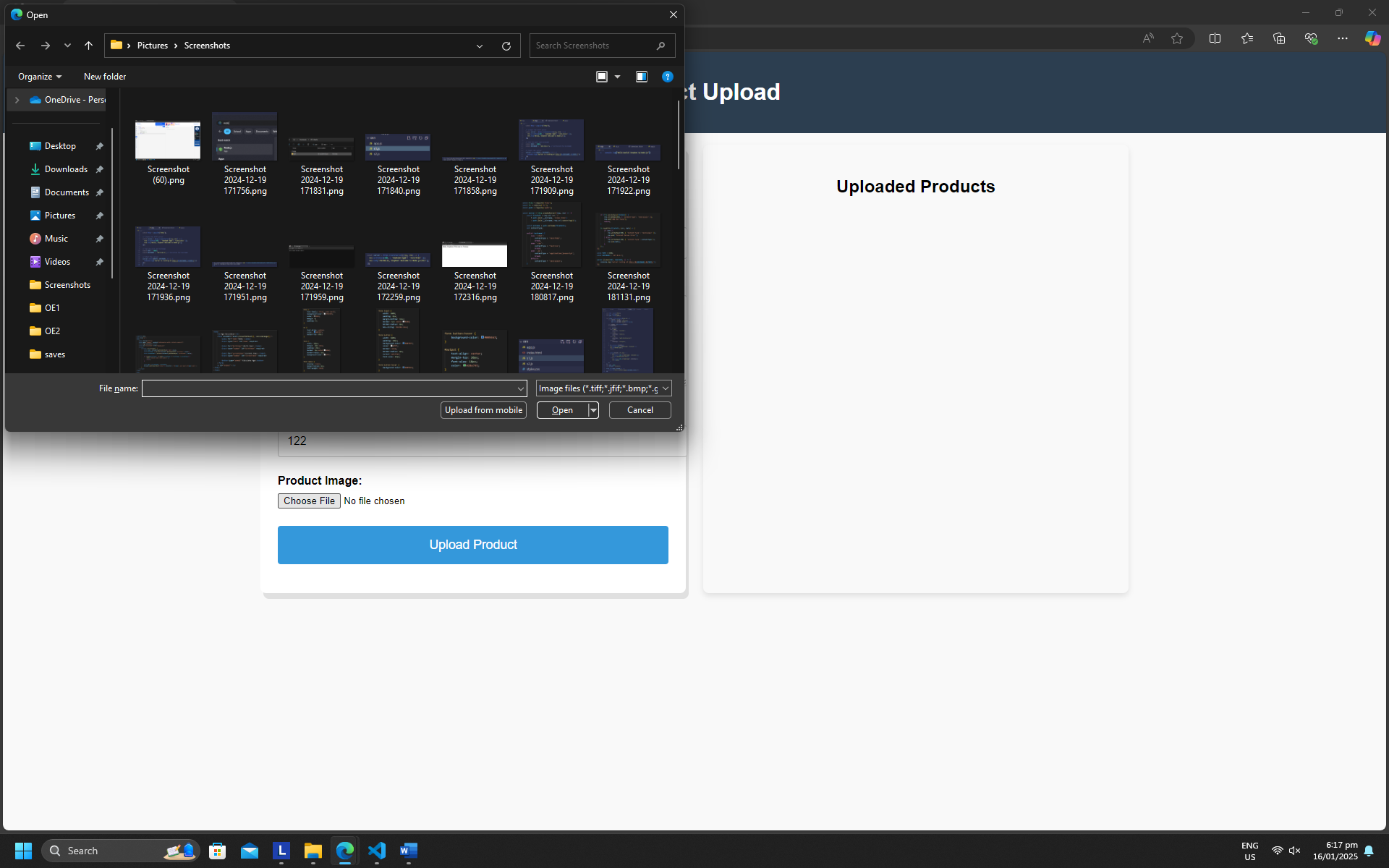
* **This is the structure of the given code, this may work if you have JS even though without a CSS since it is for design only**

***A screenshot of a computer

Description automatically generated***

***Figure 2. HTML with CSS***

* **This is the design of the structured HTML code this look okay but we will change it later on.**

******

***A screenshot of a computer

Description automatically generated***

***A screenshot of a computer

Description automatically generated***

***A screenshot of a product upload

Description automatically generated***

***Figure 3. With JS***

* **For the JS it works as the main function of the website and it also give some of the textbox and button some functionality**

After

**A screenshot of a computer

Description automatically generated**

***Figure 1. New Design***

***A screenshot of a computer screen

Description automatically generated***

***A screenshot of a computer

Description automatically generated***

***Figure 2. Functionality***

* **For the new functionality we added such as the search bar and the sorting buttons it is looking more clear user friendly**

***A screenshot of a computer

Description automatically generated***

***Figure 3. Uploading Product***

* **Once you upload the product it will be displayed in your right side, on the uploaded products. Lastly an indicator below will indicate in the bottom of the Upload Product button.**

***A screen shot of a computer

Description automatically generatedA screenshot of a product

Description automatically generated***

***Figure 4. Rating a Product***

* **For rating we make the rating star in order to make it understandable to the user on what does the start below of the product means is common in every product seller.**

***A screenshot of a computer

Description automatically generated***

***A screenshot of a shoe

Description automatically generated***

***Figure 5. Adding New Product***

* **For adding a product, it is same as the process and once you uploaded it, it will post below the first product that you uploaded**

***A screenshot of a shoe

Description automatically generated***

***A screenshot of a product

Description automatically generated***

***Figure 6. Editing Product***

* **For editing your product the name and the description of the first uploaded information will be posted in the same field and after that you can now edit the product and when done resubmit it.**

***A screenshot of a computer

Description automatically generated***

***A screenshot of a computer

Description automatically generated***

***Figure 7. Delete Product***

* **For delete there is a delete button on whatever product you want to delete select one product that you want to delete and after that a confirmation box will appear and select either of the two.**

***A screenshot of a shoe

Description automatically generated A pair of shoes with red laces

Description automatically generated***

***Figure 8. Sorting by Name***

* **For the sorting of the name, it follows the ascending so on what alphabet came first it will be the first one to order out. The left side is not yet sorted in name, and in the right side it is sorted.**

***A screenshot of a shoe

Description automatically generatedA screenshot of a shoe

Description automatically generated***

***Figure 9. Sorting by Price***

* **For the sort of price it also follows the ascending format so count it to 1 and so on, here it means that the most lowest price will come first and the last one will be the most highest price**

1. **CODES**

**Index.html**

**<!DOCTYPE *html*>**

**<html *lang*="en">**

**<head>**

**<meta *charset*="UTF-8">**

**<meta *name*="viewport" *content*="width=device-width, initial-scale=1.0"> <title>Product Upload - E-Commerce</title>**

**<link *rel*="stylesheet" *href*="styles.css">**

**</head>**

**<body>**

**<header *class*="header">**

**<h1>Product Upload</h1>**

**</header>**

**<main>**

**<section *class*="upload-form">**

**<h2>Upload New Product</h2>**

**<form *id*="productForm">**

**<div *class*="form-group">**

**<label *for*="productName">Product Name:</label>**

**<input *type*="text" *id*="productName" *placeholder*="Enter product name" *required*> </div>**

**<div *class*="form-group">**

**<label *for*="productDescription">Product Description:</label>**

**<textarea *id*="productDescription" *placeholder*="Enter product description" *required*></textarea> </div>**

**<div *class*="form-group">**

**<label *for*="productPrice">Price ($):</label>**

**<input *type*="number" *id*="productPrice" *placeholder*="Enter price" *required* *min*="0.01" *step*="0.01"> </div>**

**<div *class*="form-group">**

**<label *for*="productImage">Product Image:</label>**

**<input *type*="file" *id*="productImage" *accept*="image/\*" *required*>**

**<div *id*="imagePreview" *class*="image-preview"></div>**

**</div>**

**<button *type*="submit" *id*="submitBtn">Upload Product</button> </form>**

**<div *id*="message" *class*="message"></div>**

**</section>**

**<section *class*="product-display">**

**<h2>Uploaded Products</h2>**

**<div *id*="productList" *class*="product-list"></div>**

**</section>**

***<!-- Delete Confirmation Modal -->***

**<div *id*="deleteModal" *class*="modal" *style*="display: none;">**

**<div *class*="modal-content">**

**<p>Are you sure you want to delete this product?</p>**

**<div *class*="modal-actions">**

**<button *id*="confirmDelete" *class*="btn-confirm">Yes</button>**

**<button *id*="cancelDelete" *class*="btn-cancel">No</button>**

**</div>**

**</div>**

**</div>**

**</main>**

**<script *src*="script.js"></script>**

**</body>**

**</html>**

**Styles.css**

**body {**

**font-family: Arial, sans-serif;**

**margin: 0;**

**padding: 0;**

**background-color:**

**#e4e4e4d0;**

**}**

***.header* {**

**background-color: #da6544;**

**color: white;**

**text-align: center;**

**padding: 1rem;**

**}**

**main {**

**padding: 1rem;**

**max-width: 1200px;**

**margin: auto;**

**display: flex;**

**justify-content: space-between;**

**}**

***.upload-form* {**

**background: white;**

**padding: 1.5rem;**

**border-radius: 8px;**

**box-shadow: 4px 8px rgba(0, 0, 0, 0.1);**

**width: 45%;**

**}**

**h2 {**

**text-align: center;**

**margin-bottom: 1rem;**

**}**

***.form-group* {**

**margin-bottom: 1.5rem;**

**}**

**label {**

**display: block;**

**font-weight: bold;**

**margin-bottom: 0.5rem;**

**}**

**input[*type* = "text"], input[*type*="number"], textarea {**

**width: 95%;**

**padding: 0.75rem;**

**border: 1px solid #ccc;**

**border-radius: 4px;**

**font-size: 1rem;**

**}**

**textarea {**

**height: 100px;**

**}**

**input[*type*="file"] {**

**display: block;**

**margin-top: 0.5rem;**

**}**

**button {**

**width: 100%;**

**padding: 1rem;**

**background-color: #da6544;**

**color:white;**

**border: none;**

**border-radius: 4px;**

**font-size: 1.1rem;**

**cursor: pointer;**

**}**

**button*:hover* {**

**background-color: #df3b0d;**

**font-size: 1.3rem;**

**}**

***.image-preview* {**

**width: 100%;**

**margin-top: 1rem;**

**display: none;**

**text-align: center;**

**}**

***.image-preview* img {**

**max-width: 100%;**

**height: auto;**

**border-radius: 8px;**

**}**

***.message* {**

**margin-top: 1rem;**

**font-size: 1rem;**

**font-weight: bold;**

**color: ❑green;**

**text-align: center;**

**}**

***.product-display* {**

**width: 45%;**

**background: ■white;**

**padding: 1.5rem;**

**border-radius: 8px;**

**box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);**

**}**

***.product-list* {**

**display: flex;**

**flex-wrap: wrap;**

**gap: 1rem;**

**margin-top: 1rem;**

**}**

***.product-item* {**

**width: 100%;**

**padding: 1rem;**

**background-color: #ffffff;**

**border-radius: 8px;**

**box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);**

**}**

***.product-item* img {**

**width: 100%;**

**height: auto;**

**border-radius: 8px; }**

***.product-item* *.product-info* {**

**margin-top: 1rem;**

**}**

***.product-item* button {**

**background-color: #e74c3c;**

**border-radius: 4px;**

**margin-top: 0.5rem;**

**cursor: pointer;**

**width: 100%;**

**}**

***.product-item* button*:hover* {**

**background-color: #c0392b;**

**}**

**@media (max-width: 600px) {**

**main {**

**flex-direction: column;**

**}**

***.upload-form*, *.product-display* {**

**width: 100%;**

**}**

**}**

***.sorting* {**

**display: flex;**

**gap: 10px;**

**margin-bottom: 20px;**

**width: 550px;**

**}**

***.sorting* input {**

**padding: 5px;**

**width: 200px;**

**color:#e74c3c;**

**}**

***.rating* {**

**display: flex;**

**gap: 5px;**

**}**

***.star* {**

**cursor: pointer;**

**font-size: 1.5rem;**

**}**

***/\* for later***

***/\* Modal Styling \*/***

***.modal* {**

**position: fixed;**

**top: 0;**

**left: 0;**

**width: 100%;**

**height: 100%;**

**background-color: rgba(0, 0, 0, 0.5);**

**display: flex;**

**justify-content: center;**

**align-items: center;**

**z-index: 1000;**

**}**

***.modal-content* {**

**background: #fff;**

**padding: 20px;**

**border-radius: 8px;**

**text-align: center;**

**box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);**

**}**

***.modal-actions* {**

**margin-top: 20px;**

**}**

***.btn-confirm*, *.btn-cancel* {**

**padding: 10px 20px;**

**margin: 5px;**

**border: none;**

**border-radius: 5px;**

**cursor: pointer;**

**}**

***.btn-confirm* {**

**background-color: #e74c3c;**

**color: white;**

**}**

***.btn-confirm* button*:hover*{**

**background-color: #c0392b;**

**}**

***.btn-cancel* {**

**background-color: #e74c3c;**

**color: white;**

**}**

**Script.js**

**const productForm = document.getElementById('productForm');**

**const productName = document.getElementById('productName');**

**const productDescription = document.getElementById('productDescription');**

**const productPrice = document.getElementById('productPrice');**

**const productImage = document.getElementById('productImage');**

**const imagePreview = document.getElementById('imagePreview');**

**const message = document.getElementById('message');**

**const productList = document.getElementById('productList');**

***// Store uploaded products***

**let products = [];**

***// Show image preview***

**productImage.addEventListener('change', () => {**

**const file = productImage.files[0];**

**if (file) {**

**const reader = new FileReader();**

**reader.onload = function () {**

**imagePreview.innerHTML = `<img src="${reader.result}" alt="Product Image Preview">`;**

**imagePreview.style.display = 'block';**

**};**

**reader.readAsDataURL(file);**

**}**

**});**

**productForm.addEventListener('submit', (e) => {**

**e.preventDefault();**

**const name = productName.value.trim();**

**const description=productDescription.value.trim();**

**const price = parseFloat(productPrice.value.trim());**

**if (name === "" || description === "" || isNaN(price) || price <= 0) {**

**message.textContent = "Please fill out all fields correctly.";**

**message.style.color = "red";**

**}**

**const newProduct = {**

**id: Date.now(),**

**name,**

**description,**

**price,**

**image: imagePreview.querySelector('img').src**

**};**

**products.push(newProduct);**

**displayProducts();**

**clearForm();**

**message.textContent = "Product uploaded successfully!";**

**message.style.color = "green";**

**});**

***// Display products***

**function displayProducts() {**

**productList.innerHTML = '';**

**products.forEach (product => {**

**const productItem = document.createElement('div');**

**productItem.classList.add('product-item');**

**productItem.innerHTML = `**

**<img src="${product.image}" alt="${product.name}">**

**<div class="product-info">**

**<h3>${product.name}</h3>**

**<p>${product.description}</p>**

**<p>$${product.price}</p>**

**<button onclick="editProduct(${product.id})">Edit</button> <button onclick="deleteProduct(${product.id})">Delete</button>**

**</div>**

**`;**

**productList.appendChild(productItem);**

**});**

**}**

***// Edit product***

**function editProduct(id) {**

**const product = products.find(p => p.id === id);**

**if (product) {**

**productName.value = product.name;**

**productDescription.value = product.description;**

**productPrice.value = product.price;**

**const previewImage = document.querySelector('.image-preview img');**

**previewImage ? previewImage.src = product. image: null;**

**productImage.files = new DataTransfer().files;**

**message.textContent = "Edit the product and resubmit.";**

**message.style.color = "blue";**

**products = products.filter(p => p.id !== id);**

**displayProducts();**

**}**

**}**

***// Track the product to delete***

**let productToDelete = null;**

**function deleteProduct(id) {**

***// Set the product to delete***

**productToDelete = id;**

***// Show the modal***

**const deleteModal = document.getElementById('deleteModal');**

**deleteModal.style.display = 'flex';**

**}**

***// Handle Confirm and Cancel***

**document.getElementById('confirmDelete').addEventListener('click', () => {**

**if (productToDelete !== null) {**

***// Remove the product***

**products = products.filter(p => p.id !== productToDelete);**

***// Refresh the product list***

**displayProducts();**

***// Reset the tracker***

**productToDelete = null;**

**}**

***// Close the modal***

**document.getElementById('deleteModal').style.display = 'none';**

***// Feedback***

**alert("Product deleted successfully!");**

**});**

**document.getElementById('cancelDelete').addEventListener('click', () => {**

***// Reset the tracker***

**productToDelete = null;**

***// Close the modal***

**document.getElementById('deleteModal').style.display = 'none';**

***// Feedback***

**alert("Product deletion canceled.");**

**});**

***// Clear the form after submission***

**function clearForm() {**

**productName.value = "";**

**productDescription.value = "";**

**productPrice.value = "";**

**productImage.value = "";**

**imagePreview.style.display = 'none';**

**}**

***//For sorting***

**const sortingHTML = `**

**<div class="sorting">**

**<button id="sortByName">Sort by Name</button>**

**<button id="sortByPrice">Sort by Price</button>**

**<input type="text" id="searchBar" placeholder="Search products...">**

**</div>**

**`;**

**const productDisplay = document.querySelector('.product-display');**

**productDisplay.insertAdjacentHTML('afterbegin', sortingHTML);**

***//Sorting functionality***

**document.getElementById('sortByName').addEventListener('click', () => {**

**products.sort((a, b) => a.name.localeCompare(b.name));**

**displayProducts();**

**});**

**document.getElementById('sortByPrice').addEventListener('click', () => {**

**products.sort((a, b) => a.price - b.price);**

**displayProducts();**

**});**

***//Search Function***

**document.getElementById('searchBar').addEventListener('input', (e) => {**

**const query = e.target.value.toLowerCase();**

**const filteredProducts = products.filter(product =>**

**product.name.toLowerCase().includes(query) ||**

**product.description.toLowerCase().includes(query)**

**);**

**displayProducts(filteredProducts);**

**});**

***//Rating system***

**function displayProducts(filteredProducts = products) {**

**productList.innerHTML = '';**

**filteredProducts.forEach(product => {**

**const productItem = document.createElement('div');**

**productItem.classList.add('product-item');**

**productItem.innerHTML = `**

**<img src="${product.image}" alt="${product.name}">**

**<div class="product-info">**

**<h3>${product.name}</h3>**

**<p>${product.description}</p>**

**<p>$${product.price}</p>**

**<div class="rating" data-id="${product.id}">**

**${Array.from({ length: 5 }, (\_, i) => `<span class="star" data-value="${i + 1}">&#9734;</span>`).join('')}**

**</div>**

**<button onclick="editProduct(${product.id})">Edit</button>**

**<button onclick="deleteProduct(${product.id})">Delete</button>**

**</div>**

**`;**

**productList.appendChild(productItem);**

**});**

**setupRatings();**

**}**

**function setupRatings() {**

**document.querySelectorAll('.rating').forEach(ratingEl => {**

**const productId = parseInt(ratingEl.dataset.id);**

**const stars = ratingEl.querySelectorAll('.star');**

**stars.forEach(star => {**

**star.addEventListener('click', () => {**

**const rating = parseInt(star.dataset.value);**

**const product = products.find(p => p.id === productId);**

**product.rating = rating;**

**updateStarDisplay(ratingEl, rating);**

**alert("Thank you for the rating!")**

**});**

**});**

**const product = products.find(p => p.id === productId);**

**if (product.rating) {**

**updateStarDisplay(ratingEl, product.rating);**

**}**

**});**

**}**

**function updateStarDisplay(ratingEl, rating) {**

**const stars = ratingEl.querySelectorAll('.star');**

**stars.forEach((star, index) => {**

**star.innerHTML = index < rating ? '&#9733;' : '&#9734;';**

**});**

**}**

1. **LEARNING OUTCOMES**

* Today, we have learned how to further navigate through HTML, CSS and JS. Through this activity we had the opportunity to apply our knowledge in HTML, CSS and JS.

1. **REFERENCES**

Github

[***https://github.com/Greyriiiii/IPTL-E3.git***](https://github.com/Greyriiiii/IPTL-E3.git)