**Q2: E-Commerce Recommendation System (Binary Search Tree)**: Build an **e-commerce recommendation system** where products are stored in a **binary search tree (BST)** based on customer ratings. Implement operations to find products within a specific rating range and suggest similar products.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>E-Commerce Recommendation System</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 20px;

background-color: #f9f9f9;

}

h1 {

text-align: center;

}

.container {

max-width: 600px;

margin: auto;

padding: 20px;

background: #fff;

border-radius: 8px;

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

}

input[type="number"], button {

padding: 10px;

margin: 10px 0;

width: 100%;

border: 1px solid #ccc;

border-radius: 5px;

}

button {

background-color: #007BFF;

color: white;

font-size: 16px;

cursor: pointer;

}

button:hover {

background-color: #0056b3;

}

.results {

margin-top: 20px;

padding: 10px;

background: #e9f7ef;

border-left: 4px solid #28a745;

}

.results p {

margin: 0;

}

</style>

</head>

<body>

<div class="container">

<h1>Product Recommendation System</h1>

<label for="minRating">Minimum Rating:</label>

<input type="number" id="minRating" placeholder="Enter minimum rating (1-5)" min="1" max="5">

<label for="maxRating">Maximum Rating:</label>

<input type="number" id="maxRating" placeholder="Enter maximum rating (1-5)" min="1" max="5">

<button onclick="findProducts()">Find Products</button>

<div id="results" class="results" style="display: none;"></div>

</div>

<script>

class Product {

constructor(name, rating) {

this.name = name;

this.rating = rating;

this.left = null;

this.right = null;

}

}

class BST {

constructor() {

this.root = null;

}

insert(name, rating) {

const newProduct = new Product(name, rating);

if (!this.root) {

this.root = newProduct;

return;

}

let current = this.root;

while (true) {

if (rating < current.rating) {

if (!current.left) {

current.left = newProduct;

return;

}

current = current.left;

} else {

if (!current.right) {

current.right = newProduct;

return;

}

current = current.right;

}

}

}

findInRange(min, max, result = [], node = this.root) {

if (!node) return result;

if (node.rating >= min) this.findInRange(min, max, result, node.left);

if (node.rating >= min && node.rating <= max) result.push(`${node.name} (Rating: ${node.rating})`);

if (node.rating <= max) this.findInRange(min, max, result, node.right);

return result;

}

}

// Create the BST and add some products

const bst = new BST();

bst.insert("Laptop", 5);

bst.insert("Headphones", 4);

bst.insert("Mouse", 3);

bst.insert("Keyboard", 4);

bst.insert("Monitor", 5);

bst.insert("Phone Case", 2);

bst.insert("Charger", 1);

function findProducts() {

const minRating = parseFloat(document.getElementById("minRating").value);

const maxRating = parseFloat(document.getElementById("maxRating").value);

if (isNaN(minRating) || isNaN(maxRating) || minRating < 1 || maxRating > 5 || minRating > maxRating) {

alert("Please enter a valid rating range (1-5).");

return;

}

const results = bst.findInRange(minRating, maxRating);

const resultsDiv = document.getElementById("results");

if (results.length > 0) {

resultsDiv.innerHTML = `<p><strong>Products within rating range ${minRating}-${maxRating}:</strong></p><ul>` +

results.map(item => `<li>${item}</li>`).join('') +

`</ul>`;

} else {

resultsDiv.innerHTML = `<p>No products found in the given rating range.</p>`;

}

resultsDiv.style.display = "block";

}

</script>

</body>

</html>