

[illegible]

[illegible]

[illegible]

```

167     itemValidationMessages[itemFieldName] = {
168         required: "&nbsp;Please enter a quantity for Item " + i + "!",
169         //number: "Numbers only for Item " + i + "!",
170         //min: "Positive number only for Item " + i + "!"
171     };
172 }
173
174 // Add validation rule and message for sellerName
175 itemValidationRules["sellerName"] = {
176     required: true // Name required
177 };
178 itemValidationMessages["sellerName"] = {
179     required: "&nbsp;&nbsp;Please enter the seller's name!" // Customize message for name
180 };
181
182 $("#inputForm").validate({
183     rules: itemValidationRules,
184     messages: itemValidationMessages,
185     submitHandler: function (form) {
186         updateItemTotals();
187         sumCommission();
188         return false; // Prevent form submission
189     }
190 });
191
192
193 $("#inputForm").validate({
194     rules: itemValidationRules,
195     messages: itemValidationMessages,
196     submitHandler: function (form) {
197         updateItemTotals();
198         sumCommission();
199         return false; // Prevent form submission
200     }
201 });
202
203 // Event listener for the "CLEAR FORM" button
204 $('#clearForm').click(function () {
205     // Reset the form
206     $('#inputForm')[0].reset();
207     // Clear the calculated fields
208     $('#numField, .right-table').val('');
209 });
210
211
212 // Listen for changes in item1-4 fields
213 $(document).ready(function () {
214     $(document).tooltip();
215     for (var i = 1; i <= 4; i++) {
216         $('#item' + i).on('input', function () {
217             var itemId = $(this).data('item'); // Get the item number from
218             // data-item attribute
219             var quantity = parseFloat($(this).val()); // Parse the input value
220             // as a float
221             if (!isNaN(quantity) && quantity > 0) {
222                 // If the input is a valid positive number, update numSold1-4
223                 // with the valid quantity
224                 $('#numSold' + itemId).val(quantity);
225             } else {
226                 // If the input is not valid (negative or non-numeric), reset it
227                 // to 0
228                 $(this).val('0');
229                 $('#numSold' + itemId).val('0');
230             }
231             updateItemTotals(); // Recalculate item totals
232         });
233     }
234 });

```

```

230     });
231
232     // Function to update item totals
233     function updateItemTotals() {
234         let prices = [20.99, 12.75, 9.95, 35.89];
235         for (var i = 1; i <= 4; i++) {
236             var itemId = i; // No need to get the data-item attribute again
237             var quantity = parseFloat($('#item' + i).val());
238             if (!isNaN(quantity) && quantity >= 0) {
239                 var total = (quantity * prices[i - 1]).toFixed(2);
240                 $('#total' + i).val('$' + total);
241             } else {
242                 $('#total' + i).val('');
243             }
244         }
245     }
246
247     // Calculate and update the commission
248     function sumCommission() {
249         var totalSales = 0;
250
251         // Loop through total1 to total4 and calculate the total sales
252         for (var i = 1; i <= 4; i++) {
253             var total = parseFloat($('#total' + i).val().replace('$', '')) || 0;
254             totalSales += total;
255         }
256
257         // Calculate the commission (9% of totalSales + 250)
258         var commission = (0.09 * totalSales) + 250;
259
260         // Update the totalSold and commission for seller
261         $('#totalSold').val('$ ' + totalSales.toFixed(2)); // Add the "$" symbol
262         // before the totalSales value
263         $('#totalEarn').val('$ ' + commission.toFixed(2)); // Display with 2 decimal
264         // places
265     }
266 }
267
268 </script>
</body>
</html>

```