Soil Carbon Stock Calculator

The assessment of the amount of carbon dioxide sequestration (CO2 sequestration) in rubber plantations must focus on the analysis. Soil **Organic Carbon (SOC)** Mainly

Summarize the values that need to be sampled and analyzed in the laboratory and create a calculation model in HTML file format that can be used to calculate and generate reports systematically and easily.

1. Values that need to be analyzed in the laboratory (Lab Analysis Parameters)

To calculate the amount of carbon stored (Soil Carbon Stock, SCS), you need to know three key values at each sampling depth.

number	parameter	unit	Objective of analysis	Basic sampling/analysis methods
1	**Bulk Density (*p* b - Bulk Density)**	g/ cm3	**Most important** Used to convert concentration (%) to weight per area (tonnes /ha).	**Sampling with a soil collection cylinder (Core Sampler)** At the carbon sampling point
2	**Soil organic carbon concentration (SOC)**	% org / kg	**Most important** It is the amount of carbon stored in organic form.	**Chemical analysis** (e.g. Walkley-Black or Dry Combustion method)
3	**Gravel Content**	%	Use to adjust values *p* b If there are large stones or gravel mixed in the sample (generally size $>2\text{mm}$)	Separate and weigh in the laboratory

Random depth Soil carbon projects typically sample at standard depths, such as 0-30 cm , or in sub-divisions, such as 0-10 cm , 10-20 cm , and 20-30 cm.

2. Soil carbon calculation model for reporting

SCS calculation model is an HTML application using standard formulas based on IPCC guidelines, which calculates two main results.

1. **The amount of carbon sequestered Mg C/ha or tonnes C/ha)**
2. **CO2 gas volume stored (Mg CO2/ha)**

The formula used is

SCS (Mg C/ha) = SOC (%) x *p* b (g/cm3) x Depth(cm) x 100 x (1 - Gravel (fraction))

and CO2e (Mg/ha) = SCS x$\frac{44}{12}$ (Conversion factor ≈ 3.67)

HTML file which includes both user interface and calculations. Soil Carbon Stock Calculator)

Summary and further suggestions

This model is used to calculate the amount of carbon stored at each depth layer you sample. Once you receive your soil analysis results from the laboratory, you can take the values *p* b, SOC, and Depth (for that layer) are entered into the input fields to calculate the results immediately.

Recommendations for reporting

1. **Replication** Sampling in rubber plantations should be repeated at multiple points (e.g. 3-5 points per plantation) to obtain a more accurate average.

2. **Total Stock** If you sample multiple depth strata (e.g. 0-10, 10-20, 20-30 cm), you will need to calculate the SCS of each stratum and add them together to get the total carbon stored in the 0-30 cm depth.

3. **Comparison (Baseline)** To truly report sequestration, a similar soil analysis should be performed. **Before starting the project (Baseline)** and re-analyze **After the project started** To find the difference in carbon sequestration rate (SCS or Carbon Sequestration rate)

Example of a Soil Carbon Stock Calculator

```
<!DOCTYPE html>
<html lang="th">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>แบบจำลองการคำนวณปริมาณคาร์บอนในดิน</title>
    <script src="https://cdn.tailwindcss.com"></script>
    <style>
        @import url('https://fonts.googleapis.com/css2?family=Inter:wght@100..900&family=Noto+Sans+Thai:wgh
        body {
            font-family: 'Noto Sans Thai', 'Inter', sans-serif;
            background-color: #f7f9fc;
        }
        .card {
            box-shadow: 0 10px 15px -3px rgba(0, 0, 0, 0.1), 0 4px 6px -2px rgba(0, 0, 0, 0.05);
        }
        .result-box {
            background-color: #f0fdf4; /* Light green */
            border-left: 5px solid #10b981; /* Emerald green */
        }
    </style>
</head>
<body>

<!-- Firebase Setup (Required for Canvas Environment) -->
<script type="module">
    import { initializeApp } from "https://www.gstatic.com/firebasejs/11.6.1/firebase-app.js";
    import { getAuth, signInAnonymously, signInWithCustomToken } from "https://www.gstatic.com/firebasejs/1
    import { getFirestore, setLogLevel } from "https://www.gstatic.com/firebasejs/11.6.1/firebase-firestore

    // Global variables provided by the environment
```

```
    const appId = typeof __app_id !== 'undefined' ? __app_id : 'default-app-id';
    const firebaseConfig = typeof __firebase_config !== 'undefined' ? JSON.parse(__firebase_config) : null;
    const initialAuthToken = typeof __initial_auth_token !== 'undefined' ? __initial_auth_token : null;

    if (firebaseConfig) {
        setLogLevel('Debug');
        const app = initializeApp(firebaseConfig);
        const db = getFirestore(app);
        const auth = getAuth(app);

        // Authenticate the user
        async function authenticate() {
            try {
                if (initialAuthToken) {
                    await signInWithCustomToken(auth, initialAuthToken);
                } else {
                    await signInAnonymously(auth
                }
                // console.log("Firebase initialized and user authenticated.");
            } catch (error) {
                console.error("Firebase authentication failed:", error);
            }
        }
        authenticate();
    }
</script>

<div class="min-h-screen p-4 sm:p-8 flex items-center justify-center">
    <div class="w-full max-w-2xl bg-white p-6 sm:p-8 rounded-xl card">
        <h1 class="text-3xl font-bold text-gray-800 mb-2">แบบจำลองการคำนวณปริมาณคาร์บอนในดิน</h1>
```

Loading...

```
        <p class="text-sm text-gray-500 mb-6">ใช้สำหรับจัดทำรายงานการกักเก็บ CO₂ ในสวนยางพารา (ต่อชั้นความลึก)</p>

        <!-- Input Fields -->
        <div class="space-y-4">
            <!-- 1. Bulk Density -->
            <div>
                <label for="bulkDensity" class="block text-sm font-medium text-gray-700">1. ความหนาแน่นรวม (B
                <input type="number" id="bulkDensity" value="1.3" placeholder="เช่น 1.3" step="0.01" class="
            </div>

            <!-- 2. SOC Concentration -->
            <div>
                <label for="socPercent" class="block text-sm font-medium text-gray-700">2. ความเข้มข้นคาร์บอนอิน
                <input type="number" id="socPercent" value="1.5" placeholder="เช่น 1.5" step="0.1" class="mt-
            </div>

            <!-- 3. Layer Depth -->
            <div>
                <label for="depthCm" class="block text-sm font-medium text-gray-700">3. ความลึกของชั้นดินที่สุ่มตัวอย
                <input type="number" id="depthCm" value="30" placeholder="เช่น 30 (สำหรับ 0-30 cm)" step="1" c
            </div>

            <!-- 4. Gravel Content -->
            <div>
                <label for="gravelPercent" class="block text-sm font-medium text-gray-700">4. ปริมาณหิน/กรวด (
                <input type="number" id="gravelPercent" value="5" placeholder="เช่น 5 (ถ้าไม่มีให้ใส่ 0)" step="1"
            </div>
        </div>

        <button onclick="calculateCarbonStock()" class="mt-6 w-full py-3 px-4 border border-transparent rou
            คำนวณปริมาณคาร์บอนและ CO₂
```

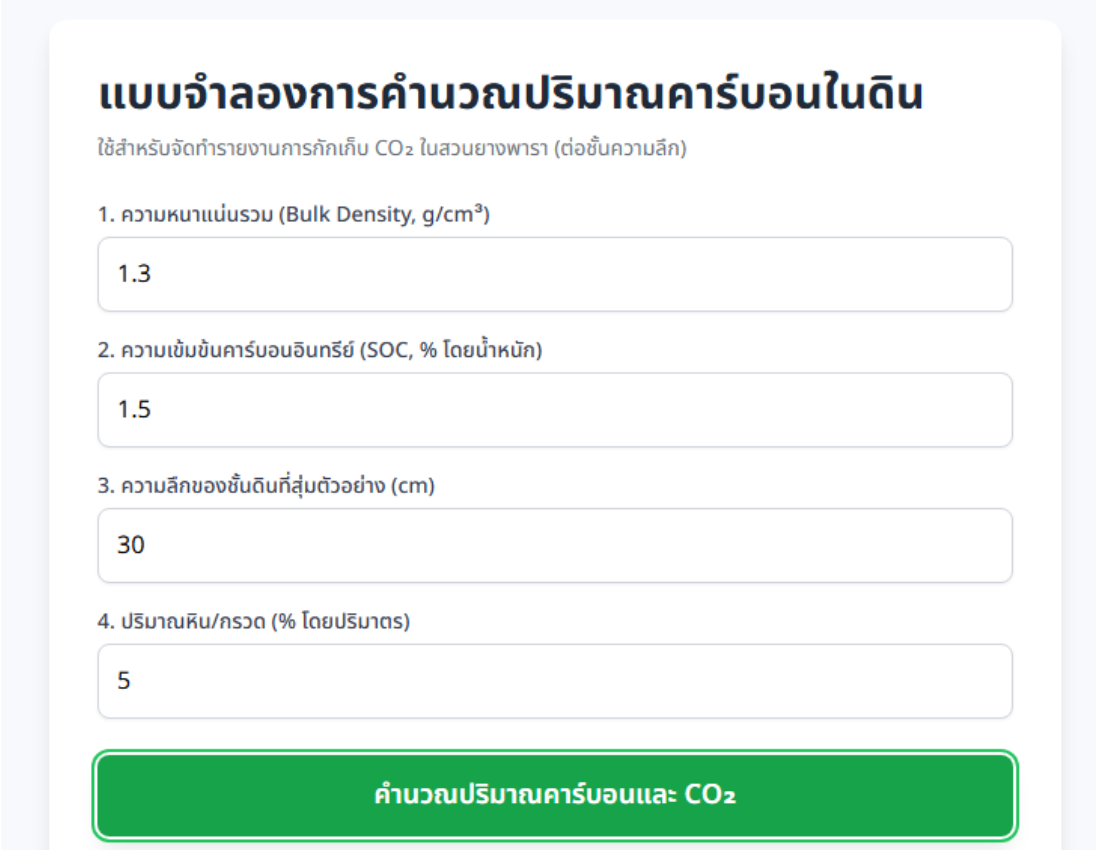
Loading...

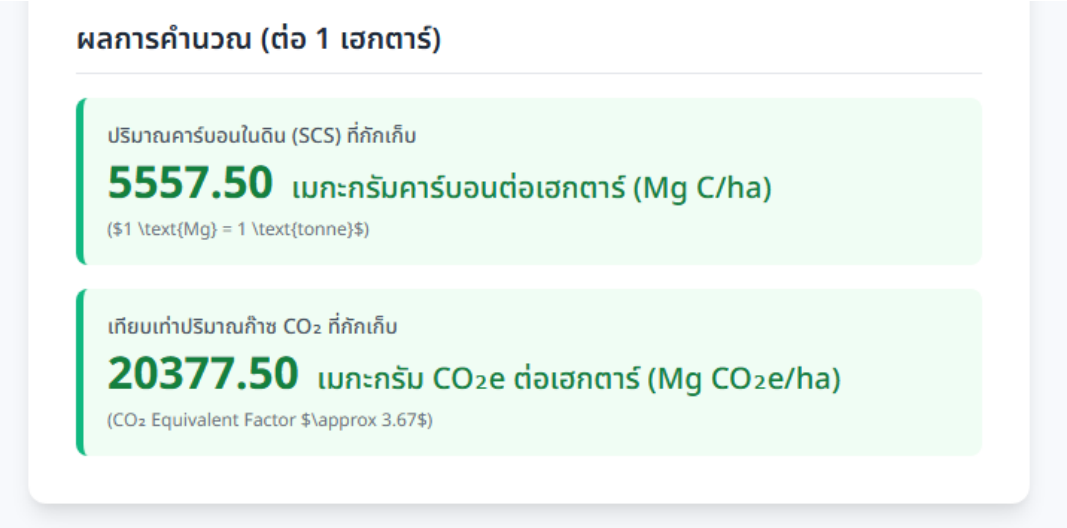
```
93          </button>
94
95          <!-- Result Display -->
96          <div id="result" class="mt-8 hidden space-y-4">
97              <h2 class="text-xl font-semibold text-gray-800 border-b pb-2">ผลการคำนวณ (ต่อ 1 เฮกตาร์)</h2>
98
99              <!-- Result 1: Carbon Stock -->
100             <div class="result-box p-4 rounded-lg">
101                 <p class="text-sm font-medium text-gray-600">ปริมาณคาร์บอนในดิน (SCS) ที่กักเก็บ</p>
102                 <p class="text-3xl font-bold text-green-700 mt-1">
103                     <span id="scsValue">0.00</span>
104                     <span class="text-xl font-medium ml-1">เมกะกรัมคาร์บอนต่อเฮกตาร์ (Mg C/ha)</span>
105                 </p>
106                 <p class="text-xs text-gray-500 mt-1">($1 \text{Mg} = 1 \text{tonne}$)</p>
107             </div>
108
109             <!-- Result 2: CO2 Equivalent -->
110             <di Loading... result-box p-4 rounded-lg">
111                 <p class="text-sm font-medium text-gray-600">เทียบเท่าปริมาณก๊าซ CO₂ ที่กักเก็บ</p>
112                 <p class="text-3xl font-bold text-green-700 mt-1">
113                     <span id="co2eValue">0.00</span>
114                     <span class="text-xl font-medium ml-1">เมกะกรัม CO₂e ต่อเฮกตาร์ (Mg CO₂e/ha)</span>
115                 </p>
116                 <p class="text-xs text-gray-500 mt-1">(CO₂ Equivalent Factor $\approx 3.67$)</p>
117             </div>
118
119             <p id="errorMsg" class="text-red-500 text-sm hidden">กรุณากรอกข้อมูลให้ครบถ้วนและถูกต้อง</p>
120         </div>
121
122     </div>
123 </div>
```

```
124
125 <script>
126     // Conversion factor to convert Carbon (C) to Carbon Dioxide Equivalent (CO2e)
127     const CO2_FACTOR = 44 / 12; // = 3.6666...
128
129     /**
130      * Calculates the Soil Carbon Stock (SCS) and CO2 Equivalent.
131      */
132     function calculateCarbonStock() {
133         // 1. Get Input Values
134         const bulkDensity = parseFloat(document.getElementById('bulkDensity').value);
135         const socPercent = parseFloat(document.getElementById('socPercent').value);
136         const depthCm = parseFloat(document.getElementById('depthCm').value);
137         const gravelPercent = parseFloat(document.getElementById('gravelPercent').value);
138
139         const resultDiv = document.getElementById('result');
140         const errorMsg = document.getElementById('errorMsg');
141
142         // 2. Validate Inputs
143         if (isNaN(bulkDensity) || isNaN(socPercent) || isNaN(depthCm) || isNaN(gravelPercent) ||
144             bulkDensity <= 0 || socPercent < 0 || depthCm <= 0 || gravelPercent < 0) {
145
146             resultDiv.classList.add('hidden');
147             errorMsg.classList.remove('hidden');
148             return;
149         }
150
151         // Hide error message if validation passes
152         errorMsg.classList.add('hidden');
153
154         // 3. Convert Gravel % to Fraction
```

```
155         const gravelFraction = gravelPercent / 100;
156
157         // 4. Calculate Soil Carbon Stock (SCS) in Mg C / ha
158         // Formula: SCS = SOC(%) * Bulk Density * Depth(cm) * 100 * (1 - Gravel Fraction)
159         // 100 is the constant factor for unit conversion (1/100 for %SOC * 10,000 m^2/ha * 1 Mg/1,000,000
160         const scsMgCPerHa = socPercent * bulkDensity * depthCm * 100 * (1 - gravelFraction);
161
162         // 5. Calculate CO2 Equivalent (CO2e) in Mg CO2e / ha
163         const co2eMgPerHa = scsMgCPerHa * CO2_FACTOR;
164
165         // 6. Display Results
166         document.getElementById('scsValue').textContent = scsMgCPerHa.toFixed(2);
167         document.getElementById('co2eValue').textContent = co2eMgPerHa.toFixed(2);
168
169         resultDiv.classList.remove('hidden');
170     }
171 </script>
172
173 </body>
174 </html>
175
```

OUTPUT





Calculation program code Soil Carbon Stock Calculator

```
<!DOCTYPE html>
<html lang="th">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title> Soil Carbon Calculation Model </title>
<script src="https://cdn.tailwindcss.com"></script>
<style>
@import url('https://fonts.googleapis.com/css2?family=Inter:wght@100..900&family=Noto+Sans+Thai:wght@100..900&display=swap');
body {
font-family: 'Noto Sans Thai', 'Inter', sans-serif;
background-color: #f7f9fc;
}
.card {
box-shadow: 0 10px 15px -3px rgba(0, 0, 0, 0.1), 0 4px 6px -2px rgba(0, 0, 0, 0.05);
}
.result-box {
background-color: #f0fdf4; /* Light green */
border-left: 5px solid #10b981; /* Emerald green */
}
</style>
</head>
<body>

<!-- Firebase Setup (Required for Canvas Environment) -->
<script type="module">
import { initializeApp } from "https://www.gstatic.com/firebasejs/11.6.1/firebase-app.js";
```

```
import { getAuth, signInAnonymously, signInWithCustomToken } from "https://www.gstatic.com/firebasejs/11.6.1/firebase-auth.js";
import { getFirestore, setLogLevel } from "https://www.gstatic.com/firebasejs/11.6.1/firebase-firestore.js";

// Global variables provided by the environment
const appId = typeof __app_id !== 'undefined' ? __app_id : 'default-app-id';
const firebaseConfig = typeof __firebase_config !== 'undefined' ? JSON.parse(__firebase_config) : null;
const initialAuthToken = typeof __initial_auth_token !== 'undefined' ? __initial_auth_token : null;

if (firebaseConfig) {
setLogLevel('Debug');
const app = initializeApp(firebaseConfig);
const db = getFirestore(app);
const auth = getAuth(app);

// Authenticate the user
async function authenticate() {
try {
if (initialAuthToken) {
await signInWithCustomToken(auth, initialAuthToken);
} else {
await signInAnonymously(auth);
}
// console.log("Firebase initialized and user authenticated.");
} catch (error) {
console.error("Firebase authentication failed:", error);
}
}
authenticate();
}
</script>

<div class="min-h-screen p-4 sm:p-8 flex items-center justify-center">
<div class="w-full max-w-2xl bg-white p-6 sm:p-8 rounded-xl card">
<h1 class="text-3xl font-bold text-gray-800 mb-2"> Soil carbon estimation model </h1>
<p class="text-sm text-gray-500 mb-6"> Used for preparing reports on CO₂ sequestration in rubber plantations (per depth layer) </p>

<!-- Input Fields -->
<div class="space-y-4">
<!-- 1. Bulk Density -->
<div>
<label for="bulkDensity" class="block text-sm font-medium text-gray-700">1. Bulk Density (g/cm³)</label>
```

```
72. <input type="number" id="bulkDensity" value="1.3" placeholder=" e.g. 1.3" step="0.01"
    class="mt-1 block w-full rounded-lg border-gray-300 shadow-sm p-3 border focus:ring-green-
    500 focus:border-green-500">
73. </div>
74.
75. <!-- 2. SOC Concentration -->
76. <div>
77. <label for="socPercent" class="block text-sm font-medium text-gray-700">2. Organic carbon
    concentration ( SOC, % by weight) </label>
78. <input type="number" id="socPercent" value="1.5" placeholder=" e.g. 1.5" step="0.1"
    class="mt-1 block w-full rounded-lg border-gray-300 shadow-sm p-3 border focus:ring-green-
    500 focus:border-green-500">
79. </div>
80.
81. <!-- 3. Layer Depth -->
82. <div>
83. <label for="depthCm" class="block text-sm font-medium text-gray-700">3. Depth of soil layer
    sampled ( cm)</label>
84. <input type="number" id="depthCm" value="30" placeholder=" e.g. 30 ( for 0-30 cm)" step="1"
    class="mt-1 block w-full rounded-lg border-gray-300 shadow-sm p-3 border focus:ring-green-
    500 focus:border-green-500">
85. </div>
86.
87. <!-- 4. Gravel Content -->
88. <div>
89. <label for="gravelPercent" class="block text-sm font-medium text-gray-700">4. Stone/gravel
    volume (% by volume) </label>
90. <input type="number" id="gravelPercent" value="5" placeholder=" e.g. 5 ( if none, enter 0)"
    step="1" class="mt-1 block w-full rounded-lg border-gray-300 shadow-sm p-3 border focus:ring-
    green-500 focus:border-green-500">
91. </div>
92. </div>
93.
94. <button onclick="calculateCarbonStock()" class="mt-6 w-full py-3 px-4 border border-
    transparent rounded-lg shadow-lg text-lg font-semibold text-white bg-green-600 hover:bg-
    green-700 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-green-500 transition
    duration-150 ease-in-out transform hover:scale-[1.01]">
95. Calculate the amount of carbon and CO₂
96. </button>
97.
98. <!-- Result Display -->
99. <div id="result" class="mt-8 hidden space-y-4">
100.        <h2 class="text-xl font-semibold text-gray-800 border-b pb-2"> Calculation results (per 1
    hectare) </h2>
101.
```

```
<!-- Result 1: Carbon Stock -->
<div class="result-box p-4 rounded-lg">
<p class="text-sm font-medium text-gray-600"> Soil Carbon Storage ( SCS) </p>
<p class="text-3xl font-bold text-green-700 mt-1">
<span id="scsValue">0.00</span>
<span class="text-xl font-medium ml-1"> Megagrams of carbon per hectare ( Mg C/ha)</span>
</p>
<p class="text-xs text-gray-500 mt-1">($1 \text{Mg} = 1 \text{tonne}$)</p>
</div>

<!-- Result 2: CO2 Equivalent -->
<div class="result-box p-4 rounded-lg">
<p class="text-sm font-medium text-gray-600"> Equivalent to the amount of CO₂ gas stored </p>
<p class="text-3xl font-bold text-green-700 mt-1">
<span id="co2eValue">0.00</span>
<span class="text-xl font-medium ml-1"> Megagrams CO₂e per hectare ( Mg CO₂e/ha)</span>
</p>
<p class="text-xs text-gray-500 mt-1">(CO₂ Equivalent Factor $\approx 3.67$)</p>
</div>

<p id="errorMsg" class="text-red-500 text-sm hidden"> Please fill in the information completely and correctly . </p>
</div>

</div>
</div>

<script>
// Conversion factor to convert Carbon (C) to Carbon Dioxide Equivalent (CO2e)
const CO2_FACTOR = 44 / 12; // = 3.6666...

/**
* Calculates the Soil Carbon Stock (SCS) and CO2 Equivalent.
*/
function calculateCarbonStock() {
// 1. Get Input Values
const bulkDensity = parseFloat(document.getElementById('bulkDensity').value);
const socPercent = parseFloat(document.getElementById('socPercent').value);
const depthCm = parseFloat(document.getElementById('depthCm').value);
const gravelPercent = parseFloat(document.getElementById('gravelPercent').value);

const resultDiv = document.getElementById('result');
const errorMsg = document.getElementById('errorMsg');
```

```
145.        // 2. Validate Inputs
146.        if (isNaN(bulkDensity) || isNaN(socPercent) || isNaN(depthCm) || isNaN(gravelPercent)
    ||
147.        bulkDensity <= 0 || socPercent < 0 || depthCm <= 0 || gravelPercent < 0) {
148.
149.        resultDiv.classList.add('hidden');
150.        errorMsg.classList.remove('hidden');
151.        return;
152.        }
153.
154.        // Hide error message if validation passes
155.        errorMsg.classList.add('hidden');
156.
157.        // 3. Convert Gravel % to Fraction
158.        const gravelFraction = gravelPercent / 100;
159.
160.        // 4. Calculate Soil Carbon Stock (SCS) in Mg C / ha
161.        // Formula: SCS = SOC(%) * Bulk Density * Depth(cm) * 100 * (1 - Gravel Fraction)
162.        // 100 is the constant factor for unit conversion (1/100 for %SOC * 10,000 m^2/ha * 1
    Mg/1,000,000 g)
163.        const scsMgCPerHa = socPercent * bulkDensity * depthCm * 100 * (1 - gravelFraction);
164.
165.        // 5. Calculate CO2 Equivalent (CO2e) in Mg CO2e / ha
166.        const co2eMgPerHa = scsMgCPerHa * CO2_FACTOR;
167.
168.        // 6. Display Results
169.        document.getElementById('scsValue').textContent = scsMgCPerHa.toFixed(2);
170.        document.getElementById('co2eValue').textContent = co2eMgPerHa.toFixed(2);
171.
172.        resultDiv.classList.remove('hidden');
173.        }
174.        </script>
175.
176.        </body>
177.        </html>
178.
```

**********end of report**********