Radioactive Decay Calculator

Table of Contents:

Chapter 1) Introduction:

1. Aim:

Aim of the project is to design a simple web-based application which aimed to calculate the radioactive decay activity of an element/s using a radioactive decay calculator.

2. Specifications:

The radioactive decay calculator has to be able to access data from a table which has been inserted by a user. The web-based calculator would provide the user with the capability to insert a nuclide name, its Half-Life, and the start activity of when the nuclide starts to decay. Both variables Half-Life and Start Activity units are set to days. After inputting the data into the table, The calculator would then calculates the nuclide decay at a different set of days and the number of days it would take to drop to less than 1 unit. Figure 1 displays the block diagram of the calculator.

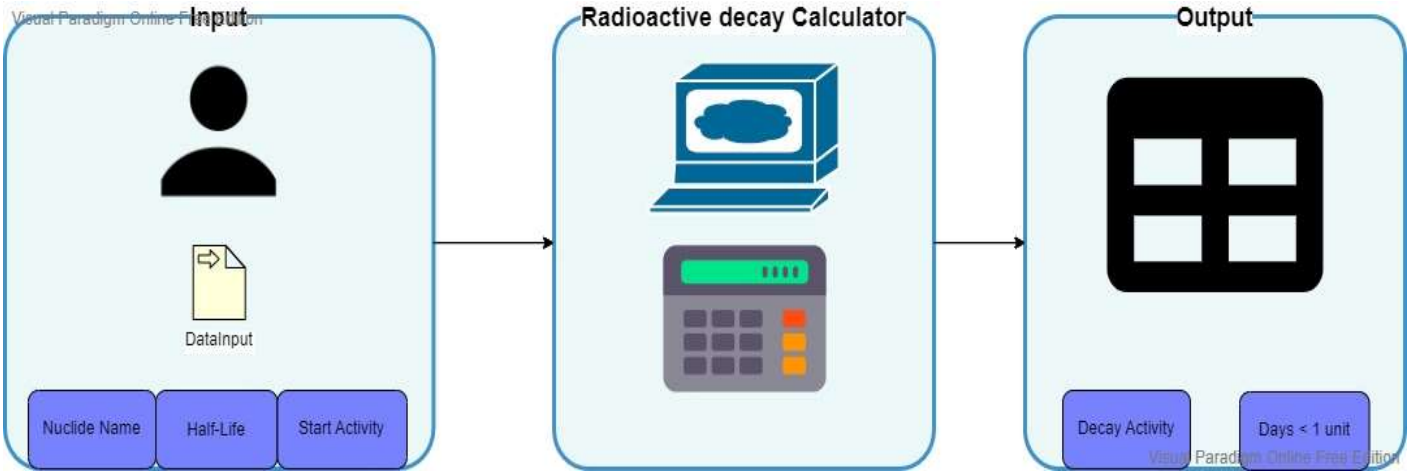

Figure 1: Block diagram of the Radioactive Decay Calculator

3. Requirements:

As the user inputs the radioactive material details, the calculator then calculates the decayed activity of the material over a certain period. The equations used to achieve the decayed activity and the days material decay to under 1 unit are found in Eq. 1.1 and Eq. 1.2,

$$Decayed\ Activity = Start\ Activity * \left(\frac{1}{2}\right)^{\frac{Time\ days}{Half-Life}} \quad \text{Eq. 1.1}$$

$$Timedays_{<1unit} = \frac{\ln\left(\frac{1}{Start\ Activity}\right) * Half - Life}{\ln\left(\frac{1}{2}\right)} \quad \text{Eq. 1.2}$$

Chapter 2) Software Design:

1. Flowchart:

To explain the approach used in creating the radioactive decay calculator, Figure 2 showcases the designed approach.

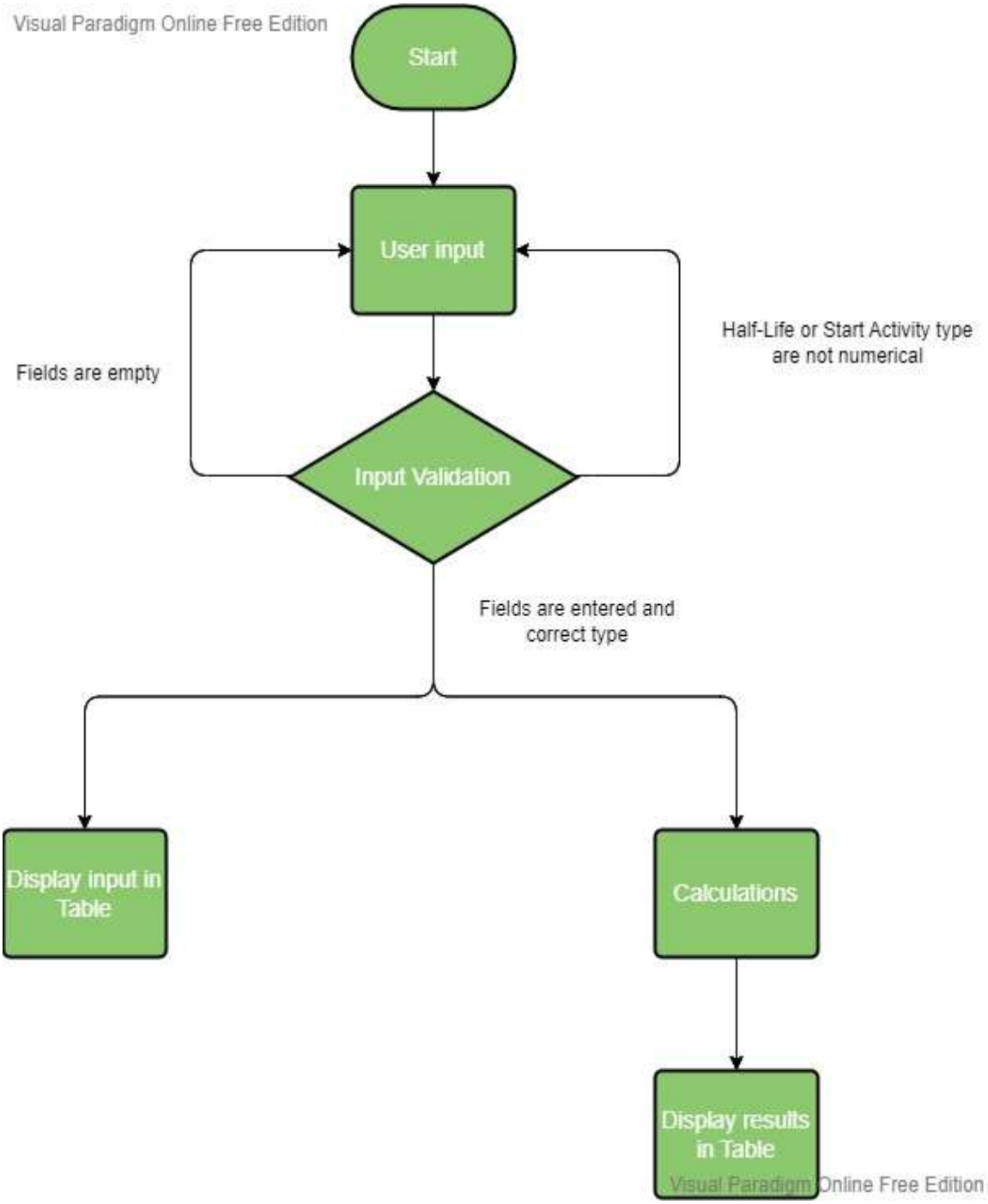

Figure 2: Flowchart diagram of the calculator

2. Coding Explanation:

The calculator is designed using JavaScript and HTML programming Language to create the Radioactive Decay Calculator. The webpage was designed to create a table with required information which is displayed in Figure 3.

Figure 3: Webpage of Radioactive Decay Calculator

This part was done by using the following code from *index.html* which is provided below,

```
<body>
        <!-- Start of the program assessment -->
        <h1 class="center">Radioactive Decay Calculator</h1>

        <!-- Creating the table to include radioactive materials -->
        <div class="container">
            <div class="tab mathalf">
                <table id="table" border="5" class="tablecenter">
                    <tr>
                        <th style="text-align: center" class="tablepadding">Nuclide name</th>
                        <th style="text-align: center" class="tablepadding">Half-Life</th>
                        <th style="text-align: center" class="tablepadding">Start Activity</th>
                        <th style="text-align: center" class="tablepadding">After 1 Day</th>
                        <th style="text-align: center" class="tablepadding">After 10 Days</th>
                        <th style="text-align: center" class="tablepadding">After 100 Days</th>
                        <th style="text-align: center" class="tablepadding">After 1000 Days</th>
                        <th style="text-align: center" class="tablepadding">Days < 1</th>
                    </tr>
                </table>
```

```
                <button id="modal-open" class="button"> Add new Nuclide ></button>

                <!-- Getting the total activity from the table -->
                <div class="center"> Time for total activity to decay:
                    <input type="number" id="totalact" disabled>
                </div>

                <div class="nuclidemodal" id="nuclide-modal">
                    <div class="nuclidemodalcontent" >
                        <div class="nuclidemodalheader">
                            <h2>Nuclide Input Data:</h2>
                            <p>Input nuclide information in the following tabs</p>
                        <!-- Creating keywords in order to relate the input variables into the
assigned keywords -->
                        <div class="tab">
                                Nuclide name: <input type="text" name="nucname" id="nucname"
placeholder="Nuclide Name">
                                Half-Life: <input type="number" name="halflife" id="halflife"
placeholder="Half-Life">
                                Start Activity: <input type="number" name="activity" id="activity"
placeholder="Start Activity">
                                <button id="submit" onclick="importdata();">Submit Data ></button>
                        </div>
                        <div class="modalclose">X</div>
                    </div>
                  </div>
                </div>
                <script src="main.js"></script>
            </div>
        </div>
    </body>
```

Code was done in order to get the input variables from the user to input radioactive material name, Half-Life and Start Activity. These variables are inserted using Modal in order to create a popup to include the information in table. Figure 4 illustrates the popup to include the information. This part was also done by using the following code from *index.html* and the user input is obtained by using the code from *main.js*. For acquiring the input data is provided below,

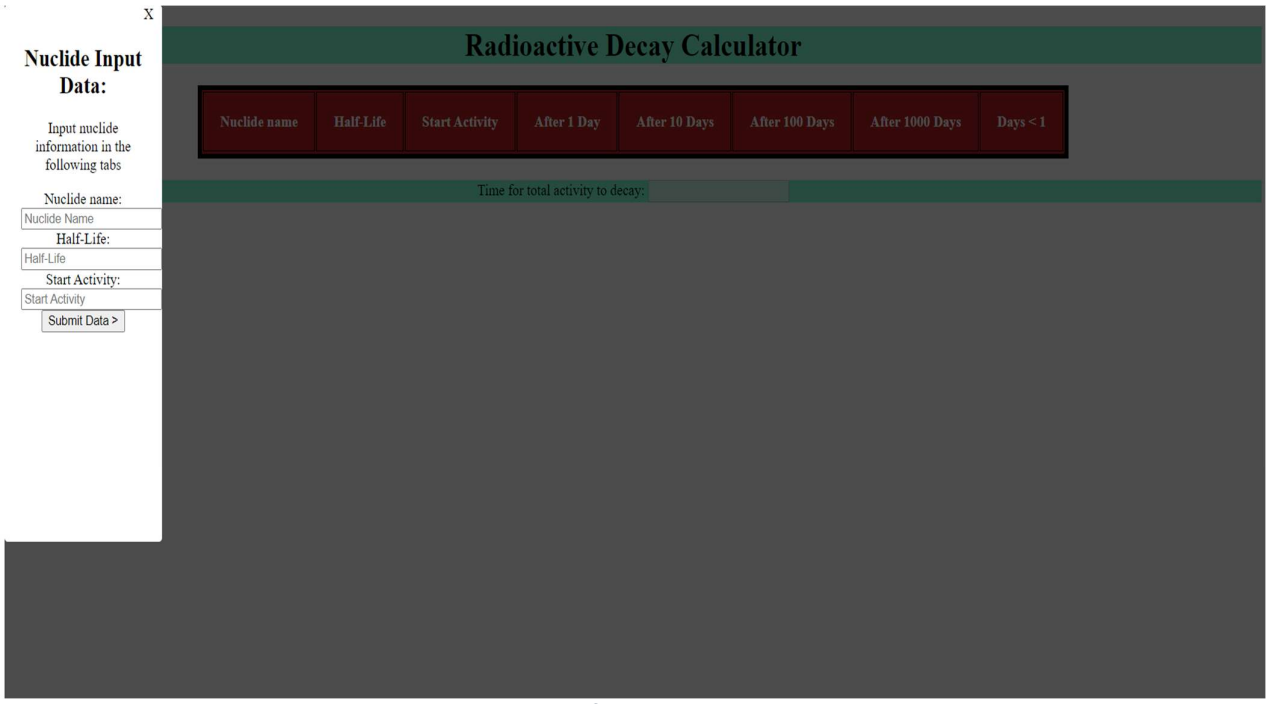

Figure 4: Popup for user input

```
// Inputing
data into
the table:
              function importdata()
              {
                  var nucname = document.getElementById('nucname').value;
                  var halflife = document.getElementById('halflife').value;
                  var activity = document.getElementById('activity').value;

                  // Checking whether the input is empty or not and if Half-Life and Activity are numbers:
                  if (!nucname || !activity || !halflife || isNaN(halflife) || isNaN(activity))
                  {
                      alert("Please fill the necessary information");
                      return;
                  }

                  var table = document.getElementById('table'),
                          NewRow = table.insertRow(table.length),
                          cell1 = NewRow.insertCell(0),
                          cell2 = NewRow.insertCell(1),
                          cell3 = NewRow.insertCell(2),
                          cell4 = NewRow.insertCell(3),
                          cell5 = NewRow.insertCell(4),
                          cell6 = NewRow.insertCell(5),
                          cell7 = NewRow.insertCell(6),
                          cell8 = NewRow.insertCell(7),
```

```
            nucname = document.getElementById('nucname').value,
            halflife = document.getElementById('halflife').value,
            activity = document.getElementById('activity').value;

    cell1.innerHTML = nucname;
    cell2.innerHTML = halflife;
    cell3.innerHTML = activity;
    cell4.innerHTML = (activity * Math.pow(0.5, (1 / halflife))).toFixed(3);
    cell5.innerHTML = (activity * Math.pow(0.5, (10 / halflife))).toFixed(3);
    cell6.innerHTML = (activity * Math.pow(0.5, (100 / halflife))).toFixed(3);
    cell7.innerHTML = (activity * Math.pow(0.5, (1000 / halflife))).toFixed(3);
    cell8.innerHTML = Math.round((((Math.log(1/activity)) * halflife) / Math.log(0.5)));
    total_activity_days()
    emptymodal();
    modal.style.display = 'none';
}
```

In the code, after obtaining the user input, it calculates the decay of the radioactive material in each column and the number of days it will take for the material to drop to less than 1 unit. Figure 5 displays the input data and calculation results with also time for total activity to decay for all materials.

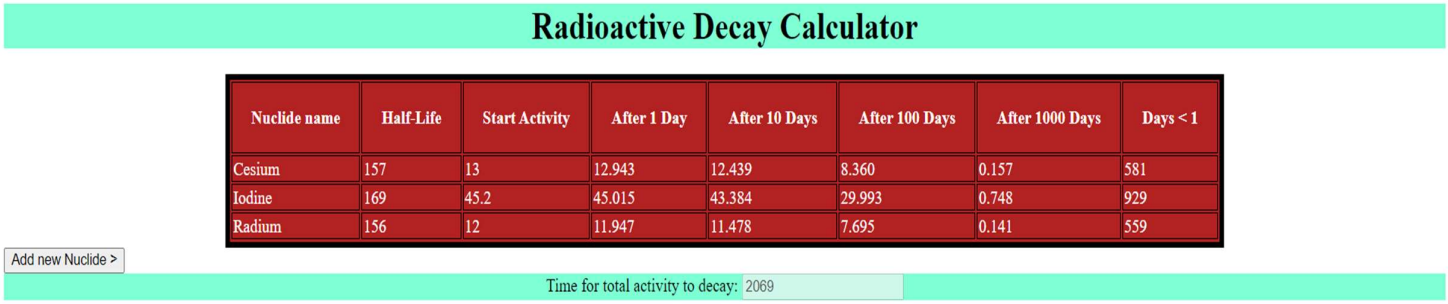

Figure 5: input data with calculation results

The time for total activity to decay for all materials can be found below,

```
//
Getting
the
total
activity
to
decay:
            function total_activity_days()
            {
                var table = document.getElementById('table');
                let total_act = 0

                for(let j = 1; j < table.rows.length; j++)
                {
                    total_act += Number(table.rows[j].cells[7].innerText);
                }

                const total_days = document.getElementById('totalact');
                total_days.value = total_act;
            }
```