**PRACTICE REPORT**

**Subject: Web application**

**Lab 2: Javascript**

*Instructor: Trần Vĩnh Khiêm*

*Report date: 22/11/2023*

1. **GENERAL INFORMATION:**

Class: MSIS207.O12.CTTT.1

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Student ID** | **Email** |
| 1 | Nguyễn Văn Trường Khoa | 20521472 | [20521472@gm.uit.edu.vn](mailto:20521472@gm.uit.edu.vn) |
| 2 |  |  |  |

## DETAILED REPORT

**Link Github contain code:**

<https://github.com/KRocker3512/Web-app-practice-class>

**Note** : If you can’t access link above, please contact to me via email :20521472@gm.uit.edu.vn .

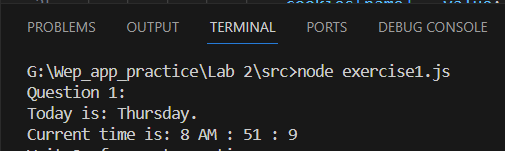
**Exercise 1:**

**Question 1:**

Code

|  |
| --- |
| console.log("Question 1:")  const currentDate = new Date();  const dayNumber = currentDate.getDay();  const days = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"];  const currentDay = days[dayNumber];  let hours = currentDate.getHours();  const minutes = currentDate.getMinutes();  const seconds = currentDate.getSeconds();  const ampm = hours >= 12 ? 'PM' : 'AM';  hours = hours % 12;  hours = hours ? hours : 12;  console.log(`Today is: ${currentDay}.`);  console.log(`Current time is: ${hours} ${ampm} : ${minutes} : ${seconds}`); |

Demo



*Image 1. Demo of exercise 1.1*.

**Question 2:**

Code

|  |
| --- |
| console.log("")          console.log("Question 3:")          function parseCookieHeader(cookieHeader) {                  const cookies = {};                  const cookieArray = cookieHeader.split(';');                  cookieArray.forEach(cookie => {                      const [name, value] = cookie.trim().split('=');                      cookies[name] = value;                  });                    return cookies;              }          const cookieHeader = "cookie1=\_hjSessionUser\_731679=eyJpZCI6IjZiYzg4NTgzLTA2ODctNWNiMi1hZmM0LTg2YTI1ODUwYmNhMCIsImNyZWF0ZWQiOjE3MDAwOTg0NjA3NDUsImV4aXN0aW5nIjpmYWxzZX0=;\_\_iid=;fpt\_uuid=%22cb599e47-590e-4896-b291-40c5cb30b2bf%22;\_hjAbsoluteSessionInProgress=0;dtdz=d596bd12-d650-4d78-89d5-912e8a1fe8c2;\_\_su=0;\_\_IP=1952847243;\_\_RC=5;ozi=2000.SSZzejyD7iu\_cVEzsr0LpYAPvhoKKa7GR9V-\_iX0Iyv-rUpesm1HYJFLvAIU2bQCEj\_hgD89M85-sUket0TOZ3So.1;undefined=undefined;\_fbp=fb.2.1700098459134.811479027;\_\_zi=3000.SSZzejyD7iu\_cVEzsr0LpYAPvhoKKa7GR9V-\_yX0Iyz-rUpfsmTGY3FIxQMI0bkADD6YfvO37umqqQofbWuUXm.1;\_\_R=3;cf\_clearance=lpmku.w0o66zPtjTxmCqwpt1.8XxPF4FjR\_nUdZPU5M-1700098453-0-1-ed57e9f4.6ae96b03.ba56b614-0.2.1700098453;\_\_uif=\_\_uid%3A1758247890457935724%7C\_\_ui%3A-1%7C\_\_create%3A1658247890;\_\_tb=0;vMobile=1;\_\_admUTMtime=1700098459;\_\_iid=;\_\_rtbh.lid=%7B%22eventType%22%3A%22lid%22%2C%22id%22%3A%22yrKoO7UZXt3kMbFEWrwB%22%7D;\_\_su=0;\_ga=GA1.3.129436683.1700098450;\_ga\_ZR815NQ85K=GS1.1.1700098449.1.1.1700098546.60.0.0;\_gcl\_au=1.1.1072636705.1700098450;\_gid=GA1.3.82745712.1700098451;\_hjFirstSeen=1;\_hjIncludedInSessionSample\_731679=0;\_hjSession\_731679=eyJpZCI6IjdmMzA2NTM3LThmZGYtNGI5YS05MDY3LTM0MWY3NGNiNDhlYSIsImNyZWF0ZWQiOjE3MDAwOTg0NjA3NDcsImluU2FtcGxlIjpmYWxzZSwic2Vzc2lvbml6ZXJCZXRhRW5hYmxlZCI6dHJ1ZX0=;\_tt\_enable\_cookie=1;\_ttp=HAsTsJC\_cH5BZVAgAGPej\_2M8LB;ajs\_group\_id=null;log\_6dd5cf4a-73f7-4a79-b6d6-b686d28583fc=cd73e63d-7b5c-44ce-a0c9-0f8821af6284";          const parsedCookies = parseCookieHeader(cookieHeader);            console.log("Parsed Cookies:", parsedCookies); |

Demo



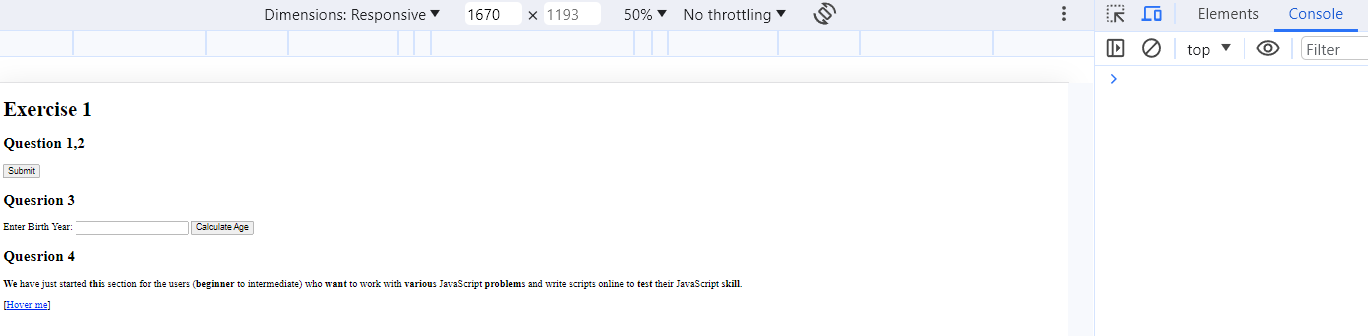
*Image 2. Demo of exercise 1.2*.

**Exercise 2:**

Code

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <title>Exercise 2</title>      <style>          .highlight {              font-weight: bold;              color: red; /\* You can customize the color \*/          }      </style>  </head>  <body>  <h1>Exercise 1</h1>  <h2>Question 1,2</h2>  <form id="myForm">      <input type="submit" value="Submit" id="submitButton">  </form>  <h2>Quesrion 3</h2>  <form id="ageForm">      <label for="birthYear">Enter Birth Year:</label>      <input type="text" id="birthYear" name="birthYear">      <button type="button" onclick="calculateAge()">Calculate Age</button>  </form>  <h2>Quesrion 4</h2>  <p><strong>We</strong> have just started <strong>this</strong> section for the users (<strong>beginner</strong> to intermediate) who <strong>want</strong> to work with <strong>various</strong> JavaScript <strong>problems</strong> and write scripts online to <strong>test</strong> their JavaScript <strong>skill</strong>.</p>  <p>[<a href="#" onMouseOver="highlight()" onMouseOut="return\_normal()">Hover me</a>]</p>  <script>      /\*Question 1\*/      document.getElementById('myForm').addEventListener('submit', function (event) {          event.preventDefault();          document.getElementById('submitButton').value = 'Submitted!';          /\*Question 2\*/          const submitButtonValue = document.getElementById('submitButton').value;          console.log('Submit Button Value:', submitButtonValue);      });      /\*Question 3\*/      function calculateAge() {          const birthYear = document.getElementById('birthYear').value;          const currentYear = new Date().getFullYear();          const age = currentYear - parseInt(birthYear);          alert('Your age is: ' + age);      }        /\*Question 4\*/      var bold\_Items;      window.onload = getBold\_items();        function getBold\_items()      {          bold\_Items = document.getElementsByTagName('strong');      }        function highlight()      {          for (var i=0; i<bold\_Items.length; i++)      {          bold\_Items[i].style.color = "red";      }      }      function return\_normal()      {          for (var i=0; i<bold\_Items.length; i++)          {              bold\_Items[i].style.color = "black";          }      }  </script>  </body>  </html> |

Demo



*Image 3. Demo of exercise 2.1 and 2.2 before click button*.

A screenshot of a computer

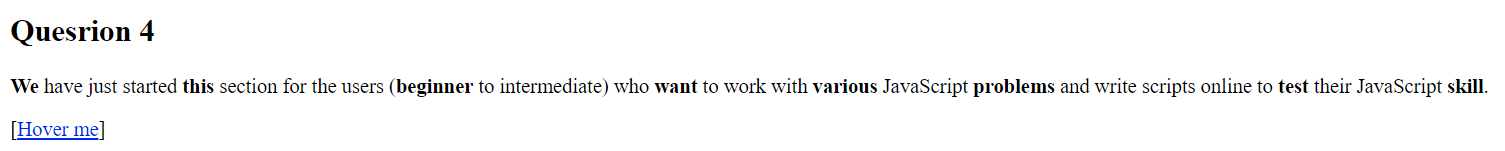
Description automatically generated

*Image 4. Demo of exercise 2.1 and 2.2 after click button*.

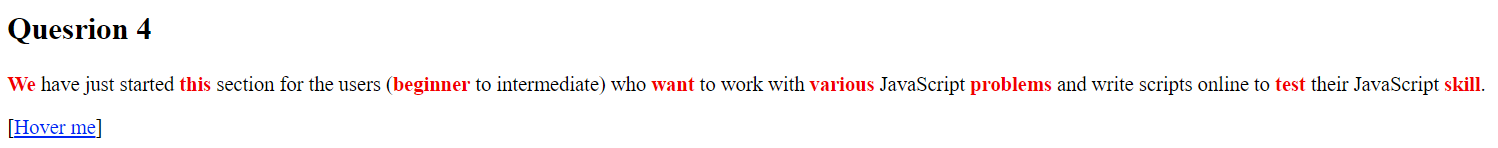
A screenshot of a computer

Description automatically generated

*Image 5. Demo of exercise 2.3 after input and click button*.



*Image 6. Demo of exercise 2.4 before hover link*.



*Image 7. Demo of exercise 2.4 after hover link*.

**Exercise 3:**

**Exercise 4:**

Cause of error link , so I can not do exercise 4

A screenshot of a computer

Description automatically generated

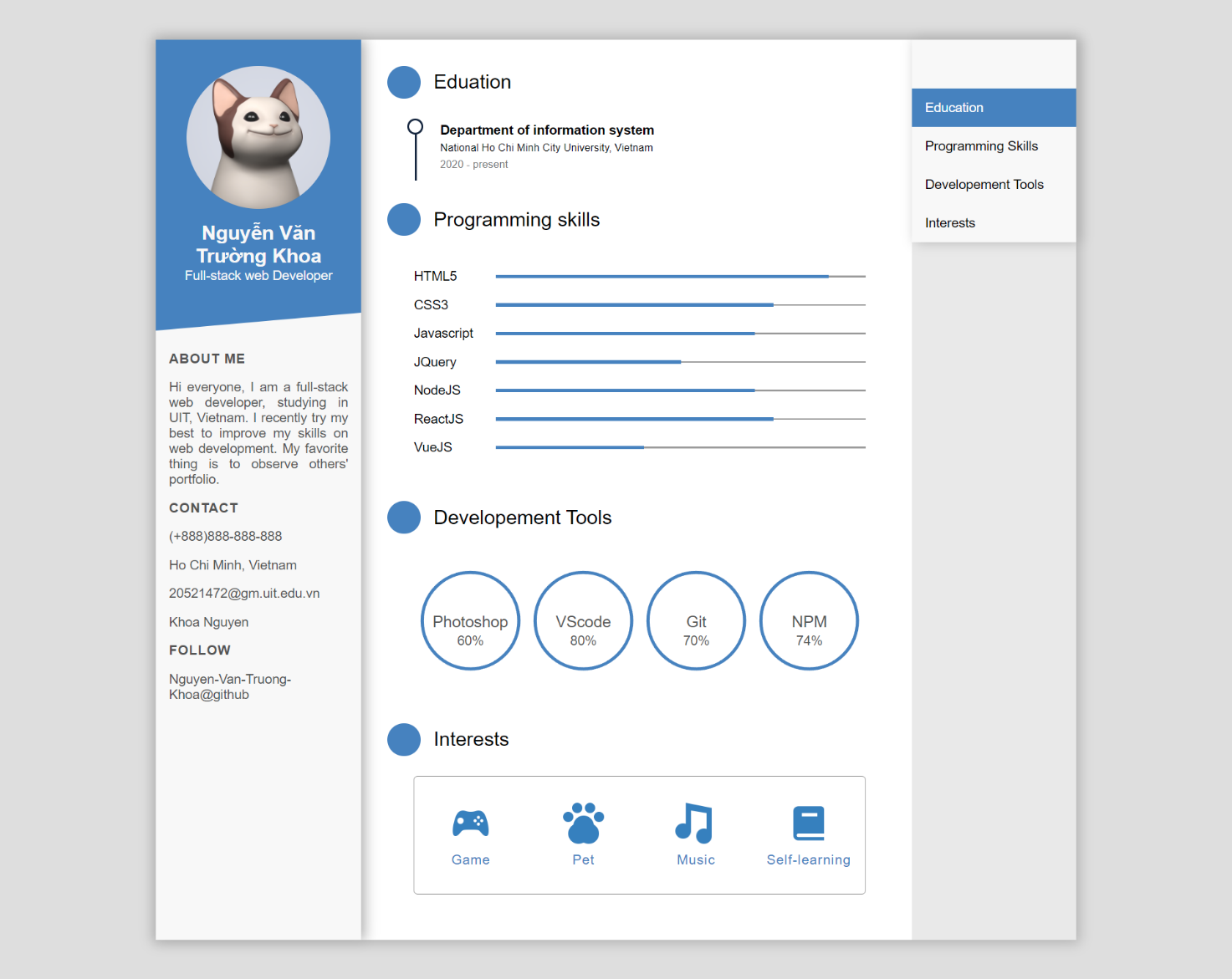
**Homework 1:**

From the old code I have like below :

A screenshot of a computer

Description automatically generated

I transfer to



Code

|  |
| --- |
|  |

|  |
| --- |
| <script>    function scrollToSection(sectionId) {      const section = document.getElementById(sectionId);      const sectionPosition = section.offsetTop;      window.scrollTo({          top: sectionPosition,          behavior: 'smooth'      });  }  </script> |

**Homework 2.1:**

A screenshot of a video game

Description automatically generated

Code

Html

|  |
| --- |
| <!DOCTYPE html>  <html lang="en" dir="ltr">  <head>    <meta charset="utf-8">    <title>Drum Kit</title>    <link rel="stylesheet" href="styles.css">    <link href="https://fonts.googleapis.com/css?family=Arvo" rel="stylesheet">  </head>  <body>    <h1 id="title">Drum 🥁 Kit</h1>    <div class="set">      <button class="w drum">w</button>      <button class="a drum">a</button>      <button class="s drum">s</button>      <button class="d drum">d</button>      <button class="j drum">j</button>      <button class="k drum">k</button>      <button class="l drum">l</button>    </div>  <script src="index.js" charset="utf-8"></script>  </body>  </html> |

Css

|  |
| --- |
| body {    text-align: center;    background-color: #283149;  }  h1 {    font-size: 5rem;    color: #DBEDF3;    font-family: "Arvo", cursive;    text-shadow: 3px 0 #DA0463;  }  footer {    color: #DBEDF3;    font-family: sans-serif;  }  .w {    background-image: url("images/tom1.png");  }  .a {    background-image: url("images/tom2.png");  }  .s {    background-image: url("images/tom3.png");  }  .d {    background-image: url("images/tom4.png");  }  .j {    background-image: url("images/snare.png");  }  .k {    background-image: url("images/crash.png");  }  .l {    background-image: url("images/kick.png");  }  .set {    margin: 10% auto;  }  .pressed {    box-shadow: 0 3px 4px 0 #DBEDF3;    opacity: 0.5;  }  .red {    color: red;  }  .drum {    outline: none;    border: 10px solid #404B69;    font-size: 5rem;    font-family: 'Arvo', cursive;    line-height: 2;    font-weight: 900;    color: #DA0463;    text-shadow: 3px 0 #DBEDF3;    border-radius: 15px;    display: inline-block;    width: 150px;    height: 150px;    text-align: center;    margin: 10px;    background-color: white;  } |

Javascript

|  |
| --- |
| var numberOfDrumButtons = document.querySelectorAll(".drum").length;  for (var i = 0; i < numberOfDrumButtons; i++) {    document.querySelectorAll(".drum")[i].addEventListener("click", function() {      var buttonInnerHTML = this.innerHTML;      makeSound(buttonInnerHTML);      buttonAnimation(buttonInnerHTML);    });  }  document.addEventListener("keypress", function(event) {    makeSound(event.key);    buttonAnimation(event.key);  });  function makeSound(key) {    switch (key) {      case "w":        var tom1 = new Audio("sounds/tom-1.mp3");        tom1.play();        break;      case "a":        var tom2 = new Audio("sounds/tom-2.mp3");        tom2.play();        break;      case "s":        var tom3 = new Audio('sounds/tom-3.mp3');        tom3.play();        break;      case "d":        var tom4 = new Audio('sounds/tom-4.mp3');        tom4.play();        break;      case "j":        var snare = new Audio('sounds/snare.mp3');        snare.play();        break;      case "k":        var crash = new Audio('sounds/crash.mp3');        crash.play();        break;      case "l":        var kick = new Audio('sounds/kick-bass.mp3');        kick.play();        break;      default: console.log(key);    }  }  function buttonAnimation(currentKey) {    var activeButton = document.querySelector("." + currentKey);    activeButton.classList.add("pressed");    setTimeout(function() {      activeButton.classList.remove("pressed");    }, 100);  } |

**Homework 2.2:**

**Problem 1 (javascript)**

|  |
| --- |
| function MakeMultiFilter(originalArray) {      let currentArray = [...originalArray];      function arrayFilterer(filterCriteria, callback) {        if (typeof filterCriteria === 'function') {          currentArray = currentArray.filter(filterCriteria);        }        if (typeof callback === 'function') {          callback.call(originalArray, currentArray);        }        return filterCriteria ? arrayFilterer : currentArray;      }      return arrayFilterer;    }      // Invoking MakeMultiFilter() with originalArray = [1, 2, 3] returns a  // function, saved in the variable arrayFilterer1, that can be used to  // repeatedly filter the input array  var arrayFilterer1 = MakeMultiFilter([1, 2, 3]);  // Call arrayFilterer1 (with a callback function) to filter out all the numbers  // not equal to 2.  arrayFilterer1(function (elem) {  return elem !== 2; // check if element is not equal to 2  }, function (currentArray) {  // 'this' within the callback function should refer to originalArray which is [1, 2, 3]  console.log(this); // prints [1, 2, 3]  console.log(currentArray); // prints [1, 3]  });  // Call arrayFilterer1 (without a callback function) to filter out all the  // elements not equal to 3.  arrayFilterer1(function (elem) {  return elem !== 3; // check if element is not equal to 3  });  // Calling arrayFilterer1 with no filterCriteria should return the currentArray.  var currentArray = arrayFilterer1();  console.log('currentArray', currentArray); // prints [1] since we filtered out 2 and 3  // Since arrayFilterer returns itself, calls can be chained  function filterTwos(elem) { return elem !== 2; }  function filterThrees(elem) { return elem !== 3; }  var arrayFilterer2 = MakeMultiFilter([1, 2, 3]);  var currentArray2 = arrayFilterer2(filterTwos)(filterThrees)();  console.log('currentArray2', currentArray2); // prints [1] since we filtered out 2 and 3  // Multiple active filters at the same time  var arrayFilterer3 = MakeMultiFilter([1, 2, 3]);  var arrayFilterer4 = MakeMultiFilter([4, 5, 6]);  console.log(arrayFilterer3(filterTwos)()); // prints [1, 3]  console.log(arrayFilterer4(filterThrees)()); // prints [4, 5, 6] |

Result

A computer screen with white text

Description automatically generated

**Problem 2 (javascript)**

|  |
| --- |
| class TemplateProcessor {      constructor(template) {        this.template = template;      }      fillIn(dictionary) {        const filledTemplate = this.template.replace(/{{\s\*([^}]+)\s\*}}/g, (match, property) => {          return dictionary[property] !== undefined ? dictionary[property] : '';        });        return filledTemplate;      }    }      var template = 'My favorite month is {{month}} but not the day {{day}} or the year {{year}}';    var dateTemplate = new TemplateProcessor(template);      var dictionary = { month: 'July', day: '1', year: '2016' };    var str = dateTemplate.fillIn(dictionary);      console.log(str); // Output: 'My favorite month is July but not the day 1 or the year 2016'      // Case: property doesn't exist in dictionary    var dictionary2 = { day: '1', year: '2016' };    var str2 = dateTemplate.fillIn(dictionary2);      console.log(str2); // Output: 'My favorite month is but not the day 1 or the year 2016' |

Result

A black background with white text

Description automatically generated

**Homework 2.3:**

Problem 1 (javascript)

|  |
| --- |
| function checkCredentials(username, password) {      if (/[^a-zA-Z\s]/.test(username) || /\s{2,}/.test(username)) {        console.log("Invalid Name or Password.");        return;      }      if (/\s/.test(password) || /[^a-zA-Z0-9@]/.test(password)) {        console.log("Invalid Name or Password.");        return;      }      console.log("Credentials are valid!");    }      // Example usage:    checkCredentials("John Doe", "Pass@123"); // Valid credentials    checkCredentials("Jane$$Doe", "Invalid Password"); // Invalid credentials |

Result

A black screen with white text

Description automatically generated

**Problem 2**

Code

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>    <meta charset="UTF-8">    <meta name="viewport" content="width=device-width, initial-scale=1.0">    <title>KIM Clock</title>    <style>      body {        display: flex;        align-items: center;        justify-content: center;        height: 100vh;        margin: 0;        font-family: 'Arial', sans-serif;        font-size: 3em;      }    </style>  </head>  <body>    <script>      function updateTime() {        const now = new Date();        const hours = now.getHours().toString().padStart(2, '0');        const minutes = now.getMinutes().toString().padStart(2, '0');        const seconds = now.getSeconds().toString().padStart(2, '0');        const timeString = `${hours}:${minutes}:${seconds}`;        document.body.textContent = timeString;      }      setInterval(updateTime, 1000);      updateTime();    </script>  </body>  </html> |

Demo

A screenshot of a computer

Description automatically generated

**Problem 3**

Code

Html

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>    <meta charset="UTF-8">    <meta name="viewport" content="width=device-width, initial-scale=1.0">    <title>New Year Countdown</title>    <link rel="stylesheet" href="styles.css">    <link href="https://fonts.googleapis.com/css2?family=Lobster&family=Roboto:wght@400;700&display=swap" rel="stylesheet">  </head>  <body>    <div class="countdown-container">      <h1 class="title">New Year Countdown</h1>      <div id="countdown" class="countdown">          <div class="background-image"></div>        <div class="countdown-item">          <div class="number" id="days"></div>          <div class="text">Days</div>        </div>        <div class="countdown-item">          <div class="number" id="hours"></div>          <div class="text">Hours</div>        </div>        <div class="countdown-item">          <div class="number" id="minutes"></div>          <div class="text">Minutes</div>        </div>        <div class="countdown-item">          <div class="number" id="seconds"></div>          <div class="text">Seconds</div>        </div>      </div>    </div>    <script src="script.js"></script>  </body>  </html> |

Css

|  |
| --- |
| body {    margin: 0;    display: flex;    align-items: center;    justify-content: center;    height: 100vh;    font-family: 'Roboto', sans-serif;    overflow: hidden;    background-image: url('./background.jfif');    background-size: cover;    background-position: center;  }  .countdown-container {    position: relative;    text-align: center;  }  .background-image {    position: absolute;    top: 35%;    left: 0;    width: 100%;    height: 70%;    background-color: rgba(255, 255, 255, 0.2);    border-radius: 5px;  }  .countdown {    font-family: 'Lobster', cursive;    font-size: 2em;    color: #fff;    display: flex;    flex-direction: row;    gap: 10px;  }  .countdown-item {    display: flex;    flex-direction: column;    align-items: center;  }  .number {    padding: 10px;    width: 50px; /\* Adjust as needed \*/    text-align: center;    color: #fff;  }  .text {    font-size: 0.5em;    color: #fff;  }  .title {    font-family: 'Lobster', cursive;    font-size: 2em;    color: #fff;  } |

Js

|  |
| --- |
| document.addEventListener('DOMContentLoaded', function () {      const daysElement = document.getElementById('days');      const hoursElement = document.getElementById('hours');      const minutesElement = document.getElementById('minutes');      const secondsElement = document.getElementById('seconds');        function updateCountdown() {        const currentTime = new Date();        const currentYear = currentTime.getFullYear();        const newYear = new Date(currentYear + 1, 0, 1);          const timeDifference = newYear - currentTime;          const days = Math.floor(timeDifference / (1000 \* 60 \* 60 \* 24));        const hours = Math.floor((timeDifference % (1000 \* 60 \* 60 \* 24)) / (1000 \* 60 \* 60));        const minutes = Math.floor((timeDifference % (1000 \* 60 \* 60)) / (1000 \* 60));        const seconds = Math.floor((timeDifference % (1000 \* 60)) / 1000);          daysElement.textContent = days;        hoursElement.textContent = hours;        minutesElement.textContent = minutes;        secondsElement.textContent = seconds;      }        setInterval(updateCountdown, 1000);        updateCountdown();    }); |

Demo

A screenshot of a computer

Description automatically generated

**Problem 4**

Code

Html

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>    <meta charset="UTF-8">    <meta name="viewport" content="width=device-width, initial-scale=1.0">    <title>Simple Calculator</title>    <link rel="stylesheet" href="styles.css">  </head>  <body>    <div class="calculator">      <input type="text" id="display" disabled>      <div class="buttons">        <button onclick="appendValue('1')">1</button>        <button onclick="appendValue('2')">2</button>        <button onclick="appendValue('3')">3</button>        <button onclick="appendOperator('+')">+</button>        <button onclick="appendValue('4')">4</button>        <button onclick="appendValue('5')">5</button>        <button onclick="appendValue('6')">6</button>        <button onclick="appendOperator('-')">-</button>        <button onclick="appendValue('7')">7</button>        <button onclick="appendValue('8')">8</button>        <button onclick="appendValue('9')">9</button>        <button onclick="appendOperator('\*')">\*</button>        <button onclick="appendValue('0')">0</button>        <button onclick="clearDisplay()">C</button>        <button onclick="calculateResult()">=</button>        <button onclick="appendOperator('/')">/</button>      </div>    </div>    <script src="script.js"></script>  </body>  </html> |

Css

|  |
| --- |
| body {      display: flex;      align-items: center;      justify-content: center;      height: 100vh;      margin: 0;      font-family: Arial, sans-serif;    }      .calculator {      text-align: center;    }      #display {      width: 200px;      margin-bottom: 10px;    }      .buttons {      display: grid;      grid-template-columns: repeat(4, 1fr);      gap: 5px;    }      button {      width: 50px;      height: 50px;      font-size: 1.2em;      cursor: pointer;    } |

Js

|  |
| --- |
| let displayValue = '';  function appendValue(value) {    displayValue += value;    updateDisplay();  }  function appendOperator(operator) {    if (displayValue !== '') {      displayValue += ` ${operator} `;      updateDisplay();    }  }  function clearDisplay() {    displayValue = '';    updateDisplay();  }  function calculateResult() {    try {      const result = eval(displayValue);      displayValue = result.toString();      updateDisplay();    } catch (error) {      displayValue = 'Error';      updateDisplay();    }  }  function updateDisplay() {    document.getElementById('display').value = displayValue;  } |

Demo

A screenshot of a computer

Description automatically generated