1. **Generate a calendar using javascript by getting the year from the user**

Program

HTML

```
<html>
 <head>
  <link rel=”stylesheet” href=”style.css”>
 </head>
 <body>
   Year:
  <input type=”text” id=”year”>
  <button onclick=”myFunction()”>submit</button>
  <div id=”calendar”></div>
  <script type=”text/javascript” src=”calendarize.js”></script>
  <script type=”text/javascript”>

      Function myFunction() {
      Var $calendar = document.getElementById(“calendar”);
      Var currentYear = document.getElementById(“year”).value;
      Var calendarize = new Calendarize();
      Calendarize.buildYearCalendar($calendar, currentYear);
      }
    </script>
 </body>
</html>
```

CSS

```
Html, body {
  Background: #e5e5e5;
  Font-family: sans-serif;
}

.month {
  Width: 300px;
  Padding: 20px;
  Background: #fff;
  Position: relative;
  Overflow: hidden;
  Float: left;
  Margin: 20px;
  Height: 350px;
}
.month h3 {
  Text-align: center;
  Margin: -20px -20px 30px -20px;
  Padding: 20px 0;
  Background: red;
  Color: #fff;
}
.day, .dow, .dummy-day {
  Display: inline-block;
  Width: 12.7864%;
  Float: left;
  Text-align: center;
  Margin-right: 1.5%;
}
.dow {
  Font-weight: bold;
  Margin-bottom: 10px;
}
.day {
  Color: #333;
  Cursor: pointer;
  Box-shadow: inset 0 0 0 1px #eee;
}
.day.weekend {
  Background: #fafaff;
}
.day:hover {
  Background: yellow;
}
.day, .dummy-day {
  Height: 40px;
  Line-height: 40px;
  Margin-bottom: 1.5%;
  Background: #fff;
}
.dummy-day {
  Background: #f5f5f5;
  Color: #ccc;
}
```

JavaScript

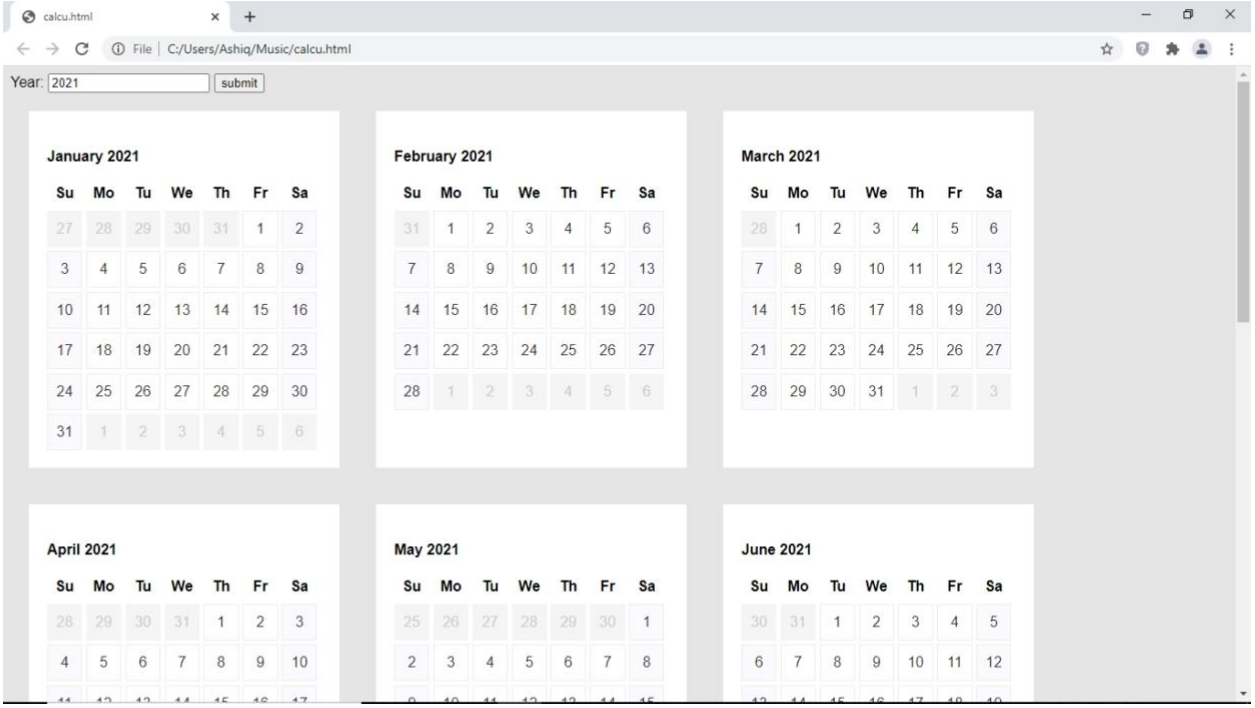
```
Function Calendarize() {
 Var monthNames = [“January”, “February”, “March”, “April”, “May”, “June”, “July”, “August”,
“September”, “October”, “November”, “December”];
 Var dayNames = [“Su”, “Mo”, “Tu”, “We”, “Th”, “Fr”, “Sa”];

 Return {

  // Return the days in a month – given a year and the month number
  getDaysInMonth: function(month, year) {
   var date = new Date(year, month, 1);
   var days = [];
   while (date.getMonth() === month) {
    days.push(new Date(date));
    date.setDate(date.getDate() + 1);
   }
   Return days;
  },

  // return an array of the first day of each month for a given year
  getMonthsInYear: function(year) {
   var date = new Date(year, 0, 1);
   var months = [];
   var monthCount = 0;
   while (monthCount < 12) {
    months.push(new Date(date));
    date.setMonth(date.getMonth() + 1);
    monthCount++;
   }
   Return months;
  },

  getMonthsInRange: function(startDate, endDate) {
   var start = new Date(startDate.getFullYear(), startDate.getMonth(), 1);
   var end = new Date(endDate.getFullYear(), endDate.getMonth(), 1);
   var months = [];
   var monthCount = 0;
   while (start <= end) {
    months.push( new Date(start) );
    start.setMonth(start.getMonth() + 1);
    monthCount++;
   }
   Return months;
```

```
},

// Create a full 12-month calendar
buildYearCalendar: function(el, year) {
  var _this = this;
  var months = _this.getMonthsInYear(year);

  var opts = {
    showMonth: true,
    showDaysOfWeek: true,
    showYear: true,
    clickHandler: function€ {
      var day = e.target.getAttribute("data-date");
      //alert(day);
    }
  };

  Months.forEach(function(a, b) {
    Var $monthNode = _this.buildMonth(b, year, opts);
    El.appendChild($monthNode);
  });
},

buildMonthsInRange: function(el, opts, startDate, limit) {
  var _this = this;
  var endDate = new Date( new Date().setDate(startDate.getDate() + limit) );
  var months = _this.getMonthsInRange(startDate, endDate);

  opts = opts  || {};
  opts.limitDate = endDate || false;
  if (opts.reverse) months = months.reverse();

  months.forEach(function(a, b) {
    var month = a.getMonth();
    var year = a.getFullYear();
    var $monthNode = _this.buildMonth(month, year, opts);
    el.appendChild($monthNode);
  });
},

// Add days and place fillers for a given month
// This function and the one above needs consolidated
buildMonth: function(monthNum, year, opts) {
  //if (monthNum === undefined || year === undefined) return "something is missing";
```

```
Var _this = this;
Var dtm = new Date(year, monthNum, 1);
Var dtmMonth = dtm.getMonth();
Var prevM = new Date(dtm.setMonth(dtmMonth – 1));
Var nextM = new Date(dtm.setMonth(dtmMonth + 1));
Var daysInMonth = _this.getDaysInMonth(monthNum, year);
Var daysPrevMonth = _this.getDaysInMonth(prevM.getMonth(), prevM.getFullYear());
Var daysNextMonth = _this.getDaysInMonth(nextM.getMonth(), nextM.getFullYear());
Var $monthNode = document.createElement(‘div’);
Var $titleNode = document.createElement(‘h4’);
Var skipLength = daysInMonth[0].getDay();
Var preLength = daysInMonth.length + skipLength;
Var postLength = function() {
  If (preLength % 7 === 0) {
    Return 0;
  } else {
    If (preLength < 35) {
      Return 35 – preLength;
    } else {
      Return 42 – preLength;
    }
  }
}

$monthNode.classList.add(‘month’);

// Add a Title to the month
If (opts.showMonth) {
  $titleNode.innerText = monthNames[monthNum] + (opts.showYear ? “ “ + year : ‘’);
  $monthNode.appendChild($titleNode);
}


// Add Days of week to the top row
If (opts.showDaysOfWeek) {
  dayNames.forEach(function(a, b) {
    var $dayNode = document.createElement(‘div’);
    $dayNode.classList.add(‘dow’);
    $dayNode.innerText = dayNames[b];
    $monthNode.appendChild($dayNode);
  });
}
```

```
// Add blank days to fill in before first day
For (var I = 0; I < skipLength; i++) {
  Var $dayNode = document.createElement(‘div’);
  $dayNode.classList.add(‘dummy-day’);
  $dayNode.innerText = daysPrevMonth.length – (skipLength – (I + 1));
  $monthNode.appendChild($dayNode);
}


// Place a day for each day of the month
daysInMonth.forEach(function(c, d) {
  var today = new Date(new Date().setHours(0, 0, 0, 0));
  var $dayNode = document.createElement(‘div’);
  $dayNode.classList.add(‘day’);
  $dayNode.setAttribute(“data-date”, c);
  $dayNode.innerText = (d + 1);
  Var dow = new Date©.getDay();
  Var dateParsed = Date.parse©;
  Var todayParsed = Date.parse(today);

  If (dateParsed === todayParsed) $dayNode.classList.add(‘today’);
  If (dateParsed > todayParsed) $dayNode.classList.add(‘future’);
  If (dateParsed <todayParsed) $dayNode.classList.add(‘past’);

  If (dow === 0 | | dow === 6) $dayNode.classList.add(‘weekend’);
  If (opts.onlyCurrent && c < today) $dayNode.classList.add(‘dummy-day’);
  If (opts.limitDate) {
    If (c > opts.limitDate) {
      $dayNode.classList.add(‘dummy-day’);
    }
  }

  If (opts.filterDayOfWeek) {
    Var valid = false;
    For (var I = 0; I < opts.filterDayOfWeek.length; i++) {
      If (c.getDay() == opts.filterDayOfWeek[i]) {
        Valid = true;
      }
    }
    If (!valid) {
      $dayNode.classList.add(‘dummy-day’);
    }
  }
  If (opts.clickHandler && !$dayNode.classList.contains(‘dummy-day’)) {
     Function handleEvent€ {
      E = e || window.event;
      e.preventDefault();
      e.stopPropagation();
      var touches = false;
      if (!touches) {
       touches = true;
       setTimeout(function() {
        touches = false;
       }, 300);
       Opts.clickHandler€;
      }
     }
     $dayNode.addEventListener("touchstart", handleEvent);
     $dayNode.addEventListener("mousedown", handleEvent);
    }
    $monthNode.appendChild($dayNode);
   });

   // Add in the dummy filler days to make an even block
   For (var j = 0; j < postLength(); j++) {
    Var $dayNode = document.createElement('div');
    $dayNode.classList.add('dummy-day');
    $dayNode.innerText = j + 1;
    $monthNode.appendChild($dayNode);
   }

   Return $monthNode;

  }
 }
}
```

OUTPUT



Year: 2021 submit

January 2021

Su	Mo	Tu	We	Th	Fr	Sa
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

February 2021

Su	Mo	Tu	We	Th	Fr	Sa
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	1	2	3	4	5	6

March 2021

Su	Mo	Tu	We	Th	Fr	Sa
28	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

April 2021

Su	Mo	Tu	We	Th	Fr	Sa
28	29	30	31	1	2	3
4	5	6	7	8	9	10

May 2021

Su	Mo	Tu	We	Th	Fr	Sa
25	26	27	28	29	30	1
2	3	4	5	6	7	8

June 2021

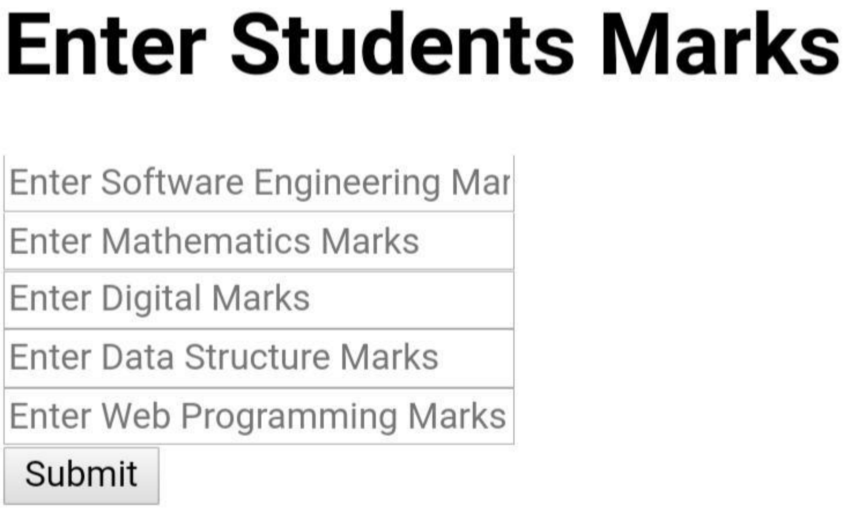
Su	Mo	Tu	We	Th	Fr	Sa
30	31	1	2	3	4	5
6	7	8	9	10	11	12

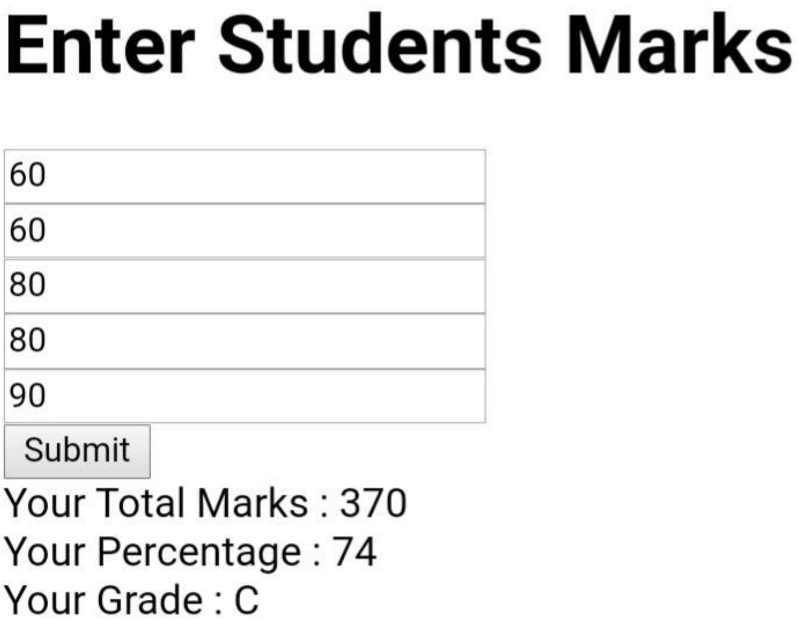
2. **Create a html page to calculate the total marks of a student by getting marks of 5 subject from the user and then show the total mark to the user**

Program

```
<!doctype html>
<html>
 <head>
  <title>Enter Marks</title>
  <script type=”text/javascript”>
Function submit_marks() {
Var sub1 = parseInt(document.getElementById(‘s1’).value);
Var sub2 = parseInt(document.getElementById(‘s2’).value);
Var sub3 = parseInt(document.getElementById(‘s3’).value);
Var sub4 = parseInt(document.getElementById(‘s4’).value);
Var sub5 = parseInt(document.getElementById(‘s5’).value);
Var total = sub1+sub2+sub3+sub4+sub5;
Var per  = total/5;
Var grade;
If (per>=35 && per<=60) {
Grade = ‘F’;
}

Else if(per>=61 && per<=70){
Grade = ‘D’;
}

Else if(per>=71 && per<=80){
Grade = ‘C’;
}

Else if(per>=81 && per<=90){
Grade = ‘B’;
}

Else if(per>=91 && per<=100){
Grade = ‘A’;
}

Else{
Grade = “Invalid or Failed”;
}
Document.getElementById(“demo”).innerHTML = “Your Total Marks : “+total+”<br>Your
Percentage : “+per+”<br>Your Grade : “+grade;

}
</script>
 </head>
 <body>
  <h1>Enter Students Marks</h1>
  <input type=”text” id=”s1” placeholder=”Enter Software Engineering Marks”>
  <br>
  <input type=”text” id=”s2” placeholder=”Enter Mathematics Marks”>
  <br>
  <input type=”text” id=”s3” placeholder=”Enter Digital Marks”>
  <br>
  <input type=”text” id=”s4” placeholder=”Enter Data Structure Marks”>
  <br>
  <input type=”text” id=”s5” placeholder=”Enter Web Programming Marks”>
  <br>
  <input type=”submit” onclick=”submit_marks()”>
  <div id=”demo”></div>
 </body>
</html>
```

OUTPUT


Enter Students Marks
Enter Software Engineering Mar
Enter Mathematics Marks
Enter Digital Marks
Enter Data Structure Marks
Enter Web Programming Marks
Submit



```
60
60
80
80
90
```

```
Your Total Marks : 370
Your Percentage : 74
Your Grade : C
```