Sunset Calculator

index.js

```
// begin cited code
// https://sunburst.sunsetwx.com/v1/docs/
// Variable created using Sunburst API from SunsetWx.
let sunburst = new SunburstJS({
  clientId: '419c8139-a5c4-4731-9d28-434f0d5bce1e',
  clientSecret: 'lxZzvOP0cJ4BoxSnhPwbQpruLFL1KHoh',
  scope: ['predictions']
});
// end cited code

// begin cited code
// https://stackoverflow.com/questions/52770661/get-latitude-and-longitude-from-zip-code-javascript
/**
 * Uses inputted zip code to find latitude, longtitude, and time zone of location and calls
 * calculate function.
 */
function calculateSunset(){
  var zip = document.querySelector("#zip").value;
  if (!validZip(zip)) {
    alert("please enter valid zip code");
  } else {
    fetch("https://maps.googleapis.com/maps/api/geocode/json?address="+zip
    +"&key=AIzaSyD-gd2vtXBRWD7GhFltpsBOBNxhRWORy-4")
    .then(response => response.json())
    .then(data => {
      var timezone = lookup(zip);
      if (timezone == null || data.results.length < 1) {
        alert("please enter valid zip code");
      } else {
        var location = data.results[0].geometry.location;
        calculate(location, timezone);
      }
    });
  }
}
// end cited code

/**
 *
 * Main method of the program that calls functions to create inputs, get data from the Sunburst
 * API, create results from the data, rank the result if the user would like to view results in
 * ranked order, and saves the results in local storage.
 *
 * @param {object} location
 * @param {string} timezone
 */
function calculate(location, timezone){
  (async () => {
    try {
      const today = new Date();
      var type = await getType();
      var days = await getDays();
      var inputs = await createInputs(today, location, type, days);

      var data = await sunburst.batchQuality(inputs);

      window.location.href = "results.html";

      var results = await createResults(today, days, timezone, data);

      if (isRanked()) {
        await rank(results);
      }

      localStorage.setItem("results", JSON.stringify(results));

    } catch (ex) {
      return console.error(ex);
    }
  })();
}

/**
 *
 * Returns boolean value containing whether or not the given zip code is in a valid format
 * (5 digits).
 *
 * @param {string} zip
 * @returns {boolean} isZip
 */
function validZip(zip) {
  const zipPattern = /^\d{5}$/;
  var isZip = zipPattern.test(zip);
  return isZip;
}

/**
 *
 * Creates and returns array of the inputs, including the location, the type (either sunrise
 * or sunset), and the dates the sunrises/sunsets occur after, given the current date, the
 * location, the type, and the array of days.
 *
 * @param {date} today
 * @param {object} location
 * @param {string} type
 * @param {int array} days
 * @returns {object array} inputs
 */
function createInputs(today, location, type, days) {
  var inputs = [];
  for (var i = 0; i < days.length; i++) {
    var date = new Date();
    inputs.push({
      geo: [location.lat, location.lng],
      type: type,
      after: date.setDate(today.getDate() + days[i]),
    });
  }
  return inputs;
}

/**
 *
 * Creates and returns array of the results, including the names of the days, the times of the
 * sunrises/sunsets, the qualities of the sunrises/sunsets, the quality percents of the
 * sunrises/sunsets, descriptions of the sunrises/sunsets, and the image paths for the
 * sunrises/sunsets, given the current date, the days selected, the time zone, and the Sunburst
 * API data.
 *
 * @param {date} today
 * @param {int array} days
 * @param {string} timezone
 * @param {sunburst batch} data
 * @returns {object array} results
 */

function createResults(today, days, timezone, data) {
  var dayNames = ["sunday", "monday", "tuesday", "wednesday", "thursday", "friday", "saturday"];
  var descriptions = ["Little to no color, with precipitation or a thick cloud layer often blocking a direct view of the sun.",
  "Some color for a short time, with conditions ranging from mostly cloudy, or hazy, to clear, with little to no clouds at all levels.",
  "A fair amount of color, often multi-colored, lasting a considerable amount of time. Often caused by scattered clouds at multiple levels.",
  "Extremely vibrant color lasting 30 minutes or more. Often caused by multiple arrangements of clouds at multiple levels, transitioning through multiple stages of vivid color."];
  var images = ["img/poor.jpg", "img/fair.jpg", "img/good.png", "img/great.jpg"];

  var results = [];
  var index = 0;
  data.forEach(({ collection, error }) => {
    if (error) {
      return console.error(error);
    } else {
      var day = new Date();
      day.setDate(today.getDate() + days[index]);

      var properties = collection.features[0].properties;

      var time = new Date(properties.validAt);
      time = time.toLocaleTimeString("en-US", {timeZone: timezone});

      var percent  = properties.qualityPercent;
      var dimgindex;
      if (percent < 25) {
        dimgindex = 0;
      } else if (percent < 50) {
        dimgindex = 1;
      } else if (percent < 75) {
        dimgindex = 2;
      } else {
        dimgindex = 3;
      }

      results.push({
        day: dayNames[day.getDay()],
        time: time,
        quality: properties.quality,
        percent: percent,
        description: descriptions[dimgindex],
        image: images[dimgindex],
      });
      index++;
    }
  });
  return results;
}

// begin cited code
// https://www.w3resource.com/javascript-exercises/searching-and-sorting-algorithm/searching-and-sorting-algorithm-exercise-4.php
/**
 *
 * Uses insertion sort to order the array of results from best sunrise/sunset quality to worst
 * sunrise/sunset quality.
 *
 * @param {object array} results
 */
function rank(results) {
  for (var i = 1; i < results.length; i++) {
    var temp = results[i];
    var j = i - 1;
    while (j >= 0 && temp.percent > results[j].percent) {
      results[j + 1] = results[j];
      j--;
    }
    results[j + 1] = temp;
  }
}
// end cited code

/**
 *
 * Returns the type (sunrise or sunset) the user would like to view based on the status of the
 * checkbox element "sunset".
 *
 * @returns {string} type
 */
function getType() {
  if (document.getElementById("sunset").checked) {
    return 'sunset';
  }
  return 'sunrise';
}

/**
 *
 * Returns an array of integers containing the value of the days relative to the current day that
 * the user has selected.
 * Ex. the previous day would be stored in the array as -1, the current day as 0, two days after
 * the current day as 2, etc.
 *
 * @returns {int array} days
 */
function getDays() {
  var days = [];
  if (document.getElementById("day-1").checked) {
    days.push(-1);
  }
  if (document.getElementById("day").checked) {
    days.push(0);
  }
  if (document.getElementById("day1").checked) {
    days.push(1);
  }
  if (document.getElementById("day2").checked) {
    days.push(2);
  }
  if (document.getElementById("day3").checked) {
    days.push(3);
  }
  if (days.length == 0) {
    alert("please choose days");
  }
  return days;
}

/**
 *
 * Returns a boolean value containing whether or not the user would like to view the sunrise
 * or sunset qualities in a ranked order.
 *
 * @returns {boolean} isRanked
 */
function isRanked() {
  var isRanked;
  if (document.getElementById("ranking").checked) {
    isRanked = true;
  } else {
    isRanked = false;
  }
  return isRanked;
}

// begin cited code
// https://www.w3schools.com/howto/howto_js_slideshow.asp
var slideIndex = 0;

/**
 *
 * Moves results slides n spaces (forward for positive values of n and backward for negative
 * values of n).
 *
 * @param {int} n
 */
function moveSlides(n) {
  slideIndex += n;
  showSlides();
}

/**
 * Displays the results slides.
 */
function showSlides() {
  var i;
  var slides = document.getElementsByClassName("slides");
  if (slideIndex >= slides.length) {
    slideIndex = 0;
  }
  if (slideIndex < 0) {
    slideIndex = slides.length - 1;
  }
  for (i = 0; i < slides.length; i++) {
      slides[i].style.display = "none";
  }
  slides[slideIndex].style.display = "block";
}
// end cited code
```

index.html

```
<!DOCTYPE html>

<html>
    <br><br>
    <head>
        <title>Sunset Calculator</title>
        <link rel="stylesheet" type="text/css" href="index.css">
        <style>
            @import url('https://fonts.googleapis.com/css2?
family=Comfortaa:wght@400;700&display=swap');
        </style>
        <script src="assets/js/sunburst.iife.js"></script>
        <script src="assets/js/zipcodes.js"></script>
        <script src="assets/js/zipcodes.json.js"></script>
        <script src="index.js"></script>
    </head>

    <body>
        <h1 class = "title">sunset calculator</h1>
        <div class = "row">
            <div class = "col-7" style = "text-align: right;">
                sunrise
            </div>

            <div class = "col-1" style = "text-align: center;">
                <label class = "switch">
                    <input type = "checkbox" id = "sunset">
                    <span class = "slider round"></span>
                </label>
            </div>

            <div class = "col-6" style = "text-align: left;">
                sunset
            </div>
        </div>

        <br>

        <div class = "row">
            <div class = "col-2"></div>
            <div class = "col-2" style = "text-align: center;">
                zip code <br>
                <input type="text" class = "zip_field" id="zip" pattern="[0-9]{5}">
            </div>

            <div class = "col-1"></div>

            <div class = "col-1" style = "text-align: center;">
                <span id = "lday-1">day-1</span>
                <input class="inp-cbx" id="day-1" type="checkbox" style="display: none"/>
                <label class="cbx" for="day-1">
                    <span>
                        <svg width="25px" height="20px" viewbox="0 0 12 10">
                            <polyline points="1.5 6 4.5 9 10.5 1"></polyline>
                        </svg>
                    </span>
                </label>
            </div>

            <div class = "col-1" style = "text-align: center;">
                <span id = "lday">day</span>
                <input class="inp-cbx" id="day" type="checkbox" style="display: none"/>
                <label class="cbx" for="day">
                    <span>
                        <svg width="25px" height="20px" viewbox="0 0 12 10">
                            <polyline points="1.5 6 4.5 9 10.5 1"></polyline>
                        </svg>
                    </span>
                </label>
            </div>

            <div class = "col-1" style = "text-align: center;">
                <span id = "lday1">day1</span>
                <input class="inp-cbx" id="day1" type="checkbox" style="display: none"/>
                <label class="cbx" for="day1">
                    <span>
                        <svg width="25px" height="20px" viewbox="0 0 12 10">
                            <polyline points="1.5 6 4.5 9 10.5 1"></polyline>
                        </svg>
                    </span>
                </label>
            </div>

            <div class = "col-1" style = "text-align: center;">
                <span id = "lday2">day2</span>
                <input class="inp-cbx" id="day2" type="checkbox" style="display: none"/>
                <label class="cbx" for="day2">
                    <span>
                        <svg width="25px" height="20px" viewbox="0 0 12 10">
                            <polyline points="1.5 6 4.5 9 10.5 1"></polyline>
                        </svg>
                    </span>
                </label>
            </div>

            <div class = "col-1" style = "text-align: center;">
                <span id = "lday3">day3</span>
                <input class="inp-cbx" id="day3" type="checkbox" style="display: none"/>
                <label class="cbx" for="day3">
                    <span>
                        <svg width="25px" height="20px" viewbox="0 0 12 10">
                            <polyline points="1.5 6 4.5 9 10.5 1"></polyline>
                        </svg>
                    </span>
                </label>
            </div>

            <div class = "col-1"></div>

            <div class = "col-1" style = "text-align: center;">
                <span id = "lranking">ranking</span>
                <input class="inp-cbx" id="ranking" type="checkbox" style="display: none"/>
                <label class="cbx" for="ranking">
                    <span>
                        <svg width="25px" height="20px" viewbox="0 0 12 10">
                            <polyline points="1.5 6 4.5 9 10.5 1"></polyline>
                        </svg>
                    </span>
                </label>
            </div>

            <div class = "col-1"></div>
        </div>

        <script language="javascript" type="text/javascript">
            var date = new Date();
            var day = (date.getDay() + 6) % 7;

            var days = ["sun", "mon", "tue", "wed", "thu", "fri", "sat"];

            document.getElementById("lday-1").innerHTML = days[day];
            document.getElementById("lday").innerHTML = days[(day+1)%7];
            document.getElementById("lday1").innerHTML = days[(day+2)%7];
            document.getElementById("lday2").innerHTML = days[(day+3)%7];
            document.getElementById("lday3").innerHTML = days[(day+4)%7];
        </script>

    <div style="text-align:center;">
        <button class="button" id="submit" onclick="calculateSunset()">submit</button>
    </div>
    </body>
</html>
```

index.css

```
.title {
  text-align: center;
  font-family: "Comfortaa", cursive;
  font-size:100px;
  letter-spacing: 10px;
  color: #0c0025;
}

body {
  font-family: "Comfortaa", cursive;
  font-size: 30px;
  color: #201141;
  background: linear-gradient(210deg, #3e59dc, #a03edc, #e75347, #ebc548, #3edcb1, #3eaddc);
  background-size: 300% 300%;
  animation: gradient 10s ease infinite;
}

@keyframes gradient {
    0% {
        background-position: 0% 50%;
    }
    50% {
        background-position: 100% 50%;
    }
    100% {
        background-position: 0% 50%;
    }
}

.col-1 {width: 6.67%;}
.col-2 {width: 13.33%;}
.col-3 {width: 20%;}
.col-4 {width: 26.67%;}
.col-5 {width: 33.33%;}
.col-6 {width: 40%;}
.col-7 {width: 46.67%;}
```

```
.col-8 {width: 53.33%;}
.col-9 {width: 60%;}
.col-10 {width: 66.67%;}
.col-11 {width: 73.33%;}
.col-12 {width: 80%;}
.col-13 {width: 86.67%;}
.col-14 {width: 93.33%}
.col-15 {width: 100%}

* {
  box-sizing: border-box;
}

[class*="col-"] {
  float: left;
  padding: 15px;
}

.row::after {
  content: "";
  clear: both;
  display: table;
}

button {
  background-color: transparent;
  font-family: "Comfortaa", cursive;
  border: 1px solid rgb(0, 47, 61);
  color: rgb(0, 47, 61);
  padding: 15px 32px;
  text-align: center;
  display: inline-block;
  font-size: 20px;
  margin: 50px;
  border-radius: 30px;
  width: 180px;
  cursor: pointer;
}

button:hover {
  background-color: #a8efff;
  border-color: #a8efff;
}

button:focus {
  outline:0;
}

/* begin cited code */
/* w3schools How To - Toggle Switch: https://www.w3schools.com/howto/howto_css_switch.asp */

/* The switch - the box around the slider */
.switch {
  position: relative;
  display: inline-block;
  width: 60px;
  height: 34px;
}

/* Hide default HTML checkbox */
.switch input {
  opacity: 0;
  width: 0;
  height: 0;
}

/* The slider */
.slider {
  position: absolute;
  cursor: pointer;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  background-color: rgb(197, 67, 67);
  -webkit-transition: .4s;
  transition: .4s;
}

.slider:before {
  position: absolute;
  content: "";
  height: 26px;
  width: 26px;
  left: 4px;
  bottom: 4px;
  background-color: white;
  -webkit-transition: .4s;
  transition: .4s;
}

input:checked + .slider {
  background-color: #2196F3;
}

input:focus + .slider {
  box-shadow: 0 0 1px #2196F3;
}

input:checked + .slider:before {
  -webkit-transform: translateX(26px);
  -ms-transform: translateX(26px);
  transform: translateX(26px);
}

/* Rounded sliders */
.slider.round {
  border-radius: 34px;
}

.slider.round:before {
  border-radius: 50%;
}

/* end cited code */



/* begin cited code */
/* Andrea Storm - checkbox css */
/* https://codepen.io/avstorm/pen/yjLGGN */

.cbx {
  margin: auto;
  -webkit-user-select: none;
  user-select: none;
  cursor: pointer;
}
.cbx span {
  display: inline-block;
  vertical-align: middle;
  transform: translate3d(0, 0, 0);
}
.cbx span:first-child {
  position: relative;
  width: 30px;
  height: 30px;
  border-radius: 3px;
  transform: scale(1);
  vertical-align: middle;
  border: 1px solid black;
  transition: all 0.2s ease;
}
.cbx span:first-child svg {
  position: absolute;
  top: 3px;
  left: 2px;
  fill: none;
  stroke: #FFFFFF;
  stroke-width: 2;
  stroke-linecap: round;
  stroke-linejoin: round;
  stroke-dasharray: 16px;
  stroke-dashoffset: 16px;
  transition: all 0.3s ease;
  transition-delay: 0.1s;
  transform: translate3d(0, 0, 0);
}
.cbx span:first-child:before {
  content: "";
  width: 100%;
  height: 100%;
  background: #a8efff;
  display: block;
  transform: scale(0);
  opacity: 1;
  border-radius: 50%;
}

.cbx:hover span:first-child {
  border-color: #a8efff;
}

.inp-cbx:checked + .cbx span:first-child {
  background: #a8efff;
  border-color: #a8efff;
  animation: wave 0.4s ease;
}
.inp-cbx:checked + .cbx span:first-child svg {
  stroke-dashoffset: 0;
}
.inp-cbx:checked + .cbx span:first-child:before {
  transform: scale(4);
  opacity: 0;
  transition: all 0.6s ease;
}

@keyframes wave {
  50% {
    transform: scale(0.9);
  }
}


.zip_field {
  font-family: inherit;
  width: 100%;
  border: 1px solid black;
  outline: 0;
  font-size: 1.3rem;
  text-align: center;
  border-radius: 25px;
  color: black;
  padding: 7px;
  background: transparent;
  transition: border-color 0.2s;
}

.zip_field:hover {
  border-color: #a8efff;
  color: #a8efff;
}

.zip_field:focus {
  padding-bottom: 6px;
  font-weight: 700;
  color: #a8efff;
  border-width: 1px;
  border-color: #a8efff;
  border-image-slice: 1;
}
.zip_field:focus ~ .zip_label {
  position: absolute;
  top: 0;
  display: block;
  transition: 0.2s;
  font-size: 1rem;
  color: #a8efff;
  font-weight: 700;
}

/* reset input */
.zip_field:required, .zip_field:invalid {
  box-shadow: none;
}

/* end cited code */
```

results.html

```
<!DOCTYPE html>

<html>
    <head>
        <title>Results</title>
        <link rel="stylesheet" type="text/css" href="results.css">
        <link rel="stylesheet" media="screen" href="https://fontlibrary.org//face/gidole-regular"
type="text/css"/>
        <style>
            @import url('https://fonts.googleapis.com/css2?
family=Comfortaa:wght@400;700&display=swap');
        </style>
        <script src="assets/js/sunburst.iife.js"></script>
        <script src="assets/js/zipcodes.js"></script>
        <script src="assets/js/zipcodes.json.js"></script>
        <script src="index.js"></script>
    </head>

    <body>
        <script language="javascript" type="text/javascript">
            var data = window.localStorage.getItem("results");
            var results = JSON.parse(data);

            for(var i = 0; i < results.length; i++) {
                document.write("<div class = 'slides'>");
                document.write("<div class = 'properties'>");
                document.write("<div class = 'day'>" + results[i].day + "</div>");
                document.write("<div class = 'time'>" + results[i].time + "</div>");
                document.write("<div class = 'quality'>" + results[i].quality + "</div>");
                document.write("<div class = 'percent'>" + results[i].percent + "%" + "</div>");
                document.write("<div class = 'description'>" + results[i].description + '</div>');
                document.write("</div>");
                document.write("<img src = " + results[i].image + ">");
                document.write("</div>");
            }
            showSlides();
        </script>
        <a class="prev" onclick="moveSlides(-1)">&#10094;</a>
        <a class="next" onclick="moveSlides(1)">&#10095;</a>
        <div style="text-align:center;">
            <button class="button" id="backhome">
                <a href="index.html">back home</a>
            </button>
        </div>
    </body>
</html>
```

results.css

```
body {
    font-family: "Comfortaa", cursive;
```

```
        color: white;
        background-color: black;
        margin: 0px;
        border: 0px;
        padding: 0px;
    }

    .day {
        top: 30%;
        font-size: 100px;
        font-weight: bold;
        letter-spacing: 40px;
        text-indent: 40px;
        padding: 15px;
    }

    .time {
        top: 40%;
    }

    .quality {
        top: 50%;
        font-weight: bold;
        padding: 15px;
    }

    .percent {
        top: 60%;
    }

    .description {
        top: 70%;
        padding: 15px 10%;
    }

    .properties {
        position: fixed;
        margin-left: auto;
        margin-right: auto;
        left: 0px;
        right: 0px;
        text-align: center;
        font-size: 30px;
        top: 12%;
    }

    img {
        width: 100%;
        height: auto;
    }

    button {
        position: fixed;
        background-color: transparent;
        font-family: "Comfortaa", cursive;
        border: 2px solid white;
        padding: 15px 22px;
        text-align: center;
        font-size: 20px;
        border-radius: 30px;
        width: 180px;
        cursor: pointer;
        position: fixed;
        margin-left: -90px;
        top: 72%;
        overflow: hidden;
    }

    button a {
        text-decoration: none;
        color: white;
    }

    button:hover {
        background-color: rgb(36, 132, 177);
        border-color: rgb(36, 132, 177);
    }

    button:focus {
        outline:0;
    }

    /* begin cited code */
    /* https://www.w3schools.com/howto/howto_js_slideshow.asp */
    * {box-sizing:border-box}

    .slides-container {
        width: 100%;
        height: auto;
        position: relative;
        margin: auto;
    }

    .slides {
      display: none;
    }

    .prev, .next {
        cursor: pointer;
        position: fixed;
        top: 50%;
        width: auto;
        margin-top: -22px;
        padding: 16px;
        color: white;
        font-weight: bold;
        font-size: 18px;
        transition: 0.6s ease;
        border-radius: 0 3px 3px 0;
        user-select: none;
      }

    .next {
        right: 0;
        border-radius: 3px 0 0 3px;
    }

    .slides {
        -webkit-animation-name: fade;
        -webkit-animation-duration: 1.5s;
        animation-name: fade;
        animation-duration: 1.5s;
    }

    @-webkit-keyframes fade {
        from {opacity: 0.5}
        to {opacity: 1}
    }

    @keyframes fade {
        from {opacity: 0.5}
        to {opacity: 1}
    }
    /* end cited code */
```