```javascript
const calendar = document.getElementById('calendar');
const tooltip = document.getElementById('tooltip');
const btn = document.querySelector("#check");
const footer = document.getElementById('scrollingText');
const header = document.getElementById('header');
const body = document.body;
let clickCount = 1;

const now = new Date();
var year = now.getFullYear();
var month = (now.getMonth() + 1);
var day = now.getDate();
var temp_name;


const dutySchedule = {
"2024-5-1": "S: 黃榮國 A: 黃煜森 N: 王瑞發 C: 孫景泰 R: 張哲維 T: 羅應順",
"2024-5-2": "S: 黃經洲 A: 唐__茂 N: 陳建中 C: 張日曜 R: 林厚運 T: 方振彬",
"2024-5-3": "S: 詹文欽 A: 王金誠 N: 許敦智 C: 秦桔萬 R: 劉錦郎 T: 呂明峰",
"2024-5-4": "S: 林森發 A: 許世勳 N: 彭偉慎 C: 柯正和 R: 黃煜森 T: 方振彬",
"2024-5-5": "S: 范振宇 A: 洪柜峰 N: 王瑞發 C: 邱冠霖 R: 余金原 T: 羅應順",
"2024-5-6": "S: 柯正和 A: 黃煜森 N: 唐__茂 C: 孫景泰 R: 劉錦郎 T: 林宏儒",
"2024-5-7": "S: 黃經洲 A: 王金誠 N: 許敦智 C: 張日曜 R: 余金原 T: 呂明峰",
"2024-5-8": "S: 林森發 A: 許世勳 N: 劉暐丞 C: 秦桔萬 R: 張哲維 T: 洪柜峰",
"2024-5-9": "S: 黃榮國 A: 劉錦郎 N: 彭偉慎 C: 邱冠霖 R: 林厚運 T: 林宏儒",
"2024-5-10": "S: 黃經洲 A: 王金誠 N: 范振宇 C: 張日曜 R: 余金原 T: 呂明峰",

};

const holiday = {

"2024-5-1": "【廿三】",
"2024-5-2": "【廿四】",
"2024-5-3": "【廿五】",
"2024-5-4": "【廿六】【放假日】",
"2024-5-5": "【立夏】【放假日】",
"2024-5-6": "【廿八】",
"2024-5-7": "【廿九】",
"2024-5-8": "【四月小】",
"2024-5-9": "【初二】",
"2024-5-10": "【初三】",
};


function Zellercongruence(day, month, year)
{
        if (month == 1)
        {
            month = 13;
            year--;
        }
        if (month == 2)
        {
            month = 14;
            year--;
        }
        let q = day;
        let m = month;
        let k = year % 100;
        let j = parseInt(year / 100, 10);
        let h = q + parseInt(13 * (m + 1) / 5, 10) + k + parseInt(k / 4, 10) + parseInt(j / 4, 10) + 5 * j;
        h = h % 7;
        switch (h)
        {
            case 0:
              return 6;
              break;
            case 1:
                return 0;
                break;
            case 2:
                return 1;
                break;
            case 3:
                return 2;
                break;
            case 4:
                return 3;
                break;
            case 5:
                return 4;
                break;
            case 6:
                return 5;
                break;
        }
}

function addEventListeners(dayElement, btn, day, month, year, date) {
  dayElement.addEventListener('mouseover', () => {
    let todays = document.querySelectorAll('.today');
      todays.forEach(today => {
        today.style.backgroundColor = '';
      });

    if (btn.checked) {
      dayElement.style.color = 'purple'; // Change the color when button is checked and mouse is over
    }
    showTooltip(date);

  });

  dayElement.addEventListener('mouseout', () => {
    if (btn.checked) {
      dayElement.style.color = 'white'; // Reset the color when button is checked and mouse is out
      let today = document.querySelectorAll('.today');
      today.forEach(today => {
        today.style.color = 'MediumBlue';
      });
      if (Zellercongruence(day, month, year) === 0 || Zellercongruence(day, month, year) === 6) {
        dayElement.style.color = 'yellow';
      }
    }
    hideTooltip();
  });
}
function addEventListener_toHideToolTipandShowToday(headerCell) {
  headerCell.addEventListener('click', () => {
    const formattedDate = `${year}-${month}-${day}`
    if (clickCount % 2 === 0) {
      hideTooltip();
      let todays = document.querySelectorAll('.today');
      todays.forEach(today => {
        today.style.backgroundColor = '';
      });
      // Perform actions for hiding tooltip
    } else {
      let todays = document.querySelectorAll('.today');
      todays.forEach(today => {
        today.style.backgroundColor = '#ffff99';
      });
      showTooltip(formattedDate);
      if (btn.checked){
        hideTooltip();
      }
    }
    clickCount++;
  });

}

// Function to create calendar days
```

```javascript
function createCalendar(year, month) {
  const daysInMonth = new Date(year, month, 0).getDate();
  const weekdays = ['一', '二', '三', '四', '五', '六', '日'];
  var counter = 0;
  var counterN = 0;
  var now = new Date();
  var year_now = now.getFullYear();
  var month_now = (now.getMonth() + 1); // Months are zero-based
  var day_now = now.getDate();
  var yearN;
  var nextMonth = (now.getMonth() + 1) + 1;
    if (nextMonth > 12) {
      nextMonth = 1;
      yearN = year + 1;
    }
  yearN=year;
  const daysInMonthN = new Date(yearN, nextMonth, 0).getDate();
  dayOfWeekN = Zellercongruence(1, nextMonth, yearN);
  if (dayOfWeekN === 1) {
    counterN=0;
  } else if (dayOfWeekN === 2) {
    counterN=1;
  } else if (dayOfWeekN === 3) {
    counterN=2;
  } else if (dayOfWeekN === 4) {
    counterN=3;
  } else if (dayOfWeekN === 5) {
    counterN=4;
  } else if (dayOfWeekN === 6) {
    counterN=5;
  } else {
    counterN=6;
  }
  const headerCell = document.createElement('div');
  headerCell.classList.add('header-cell');
  headerCell.textContent = ` ${year} 年IDSP${month} 月  `;
  header.appendChild(headerCell);
  addEventListener_toHideToolTipandShowToday(headerCell);


  const date = `${year}-${month}-${day}`;

  for (let i=0; i < 7; i++){
    const weekdayElement = document.createElement('div');
    weekdayElement.classList.add('weekday');
    weekdayElement.textContent = weekdays[i];
    calendar.appendChild(weekdayElement);
  }

    dayOfWeek = Zellercongruence(1, month, year);

    if (dayOfWeek === 1) {
      counter=0;
    } else if (dayOfWeek === 2) {
      counter=1;
      for (let i = 0; i < counter; i++) {
        const dayElement = document.createElement('div');
        dayElement.classList.add('day');
        dayElement.textContent = "";
        calendar.appendChild(dayElement);
      }
    } else if (dayOfWeek === 3) {
      counter=2;
      for (let i = 0; i < counter; i++) {
        const dayElement = document.createElement('div');
        dayElement.classList.add('day');
        dayElement.textContent = "";
        calendar.appendChild(dayElement);
      }
    } else if (dayOfWeek === 4) {
      counter=3;
      for (let i = 0; i < counter; i++) {
        const dayElement = document.createElement('div');
        dayElement.classList.add('day');
        dayElement.textContent = "";
        calendar.appendChild(dayElement);
      }
    } else if (dayOfWeek === 5) {
      counter=4;
      for (let i = 0; i < counter; i++) {
        const dayElement = document.createElement('div');
        dayElement.classList.add('day');
        dayElement.textContent = "";
        calendar.appendChild(dayElement);
      }
    } else if (dayOfWeek === 6) {
      counter=5;
      for (let i = 0; i < counter; i++) {
        const dayElement = document.createElement('div');
        dayElement.classList.add('day');
        dayElement.textContent = "";
        calendar.appendChild(dayElement);
      }
    } else {
      counter=6;
      for (let i = 0; i < counter; i++) {
        const dayElement = document.createElement('div');
        dayElement.classList.add('day');
        dayElement.textContent = "";
        calendar.appendChild(dayElement);
      }
    }

  for (let day = 1; day <= daysInMonth; day++) {
    const date = `${year}-${month}-${day}`;
    const dayElement = document.createElement('div');
    dayElement.classList.add('day');
    dayElement.textContent = day;
    calendar.appendChild(dayElement);

    if (Zellercongruence(day, month, year) === 0 || Zellercongruence(day, month, year) === 6) {
      dayElement.classList.add('weekend');
    }
    if (year === year_now && month === month_now && day === day_now) {
      dayElement.classList.add('today');
    }
    addEventListeners(dayElement, btn, day, month, year, date);
  }

  //below is created for the next month. If the days of next month exceeds this month
  for (let i = 0; i < (daysInMonthN + counterN - daysInMonth - counter) ; i++) {
    const dayElement = document.createElement('div');
    dayElement.classList.add('day');
    dayElement.textContent = "";
    calendar.appendChild(dayElement);
  }

}

function showTooltip(date) {
  const calendarRect = calendar.getBoundingClientRect();

  tooltip.style.left = `${calendarRect.left-102}px`;
  tooltip.style.top = `${calendarRect.top+106}px`;

  tooltip.textContent = dutySchedule[date] || "None";
  tooltip.style.display = 'block';

  document.title = holiday[date] || "None";
}



function hideTooltip() {
  tooltip.style.display = 'none';
  document.title = 'On Duty Calendar';
}

function change() {

  var day_now = now.getDate();
  if (btn.checked) {
    updateSelection();
    clearSelectedClass();
    hideTooltip();

    let todays = document.querySelectorAll('.today');
      todays.forEach(today => {
        today.style.backgroundColor = '';
      });

    let previousToday = document.querySelectorAll('.today');
    previousToday.forEach(today => {
      today.classList.remove('today');
    });

    body.classList.add("dark");
    body.style.backgroundImage = "url('TIAC.png')";

    let days = document.querySelectorAll('.day');
    days.forEach(day => {
      day.style.color = 'white';
    });

    let weekendDays = document.querySelectorAll('.weekend');
    weekendDays.forEach(day => {
      day.style.color = 'yellow';
    });

    let header = document.getElementById('header');
    header.style.color = 'white';
    footer.style.color = 'MediumBlue';



    month = (now.getMonth() + 1) + 1;
    if (month > 12) {
      month = 1;
      year++;
    }
    var daysInNextMonth = new Date(year, month, 0).getDate();
    days.forEach((day, index) => {
      day.textContent = "";
      if (index >= Zellercongruence(1, month, year) - 1 && index < daysInNextMonth + Zellercongruence(1, month, year) - 1) {
        const date = `${year}-${month}-${index - Zellercongruence(1, month, year) + 2}`;
        day.textContent = index - Zellercongruence(1, month, year) + 2;
        addEventListeners(day, btn, index - Zellercongruence(1, month, year) + 2, month, year, date);
      }else{
        addEventListeners(day, btn, index - Zellercongruence(1, month, year) + 2, month, year, 999);
      }
    });
    const headerCell = document.querySelector('.header-cell');
    headerCell.textContent = ` ${year} 年IDSP${month} 月  `;
    highlightAdditionalHoliday();
  } else {
    updateSelection();
    clearSelectedClass();

    body.classList.remove("dark");
    body.style.backgroundImage = "url('tower.png')";

    // Reset the color of '.day' elements to their default
    let days = document.querySelectorAll('.day');
    days.forEach(day => {
      day.style.color = ''; // This will remove the inline 'color' style, allowing the CSS rule to take effect
    });
    // Reset the color of 'header' element to its default
    let header = document.getElementById('header');
    header.style.color = ''; // This will remove the inline 'color' style, allowing the CSS rule to take effect
    footer.style.color = 'black';

    month = (now.getMonth() + 1);
    if (month === 12) {
      year--;
    }
    var daysInCurrentMonth = new Date(year, month, 0).getDate();
    days.forEach((day, index) => {
      day.textContent = "";
      if (index >= Zellercongruence(1, month, year) - 1 && index < daysInCurrentMonth + Zellercongruence(1, month, year) - 1) {
        const date = `${year}-${month}-${index - Zellercongruence(1, month, year) + 2}`;
        day.textContent = index - Zellercongruence(1, month, year) + 2;
        addEventListeners(day, btn, index - Zellercongruence(1, month, year) + 2, month, year, date);
      }else {
        addEventListeners(day, btn, index - Zellercongruence(1, month, year) + 2, month, year, 999);
      }
      if (index - Zellercongruence(1, month, year) + 2 === day_now) {
        day.classList.add('today');
      }
    });

    const headerCell = document.querySelector('.header-cell');
    headerCell.textContent = ` ${year} 年IDSP${month} 月  `;
    fetchWeather();
  }
  //below is to highlight the name previously selected in the change function
  highlightSelectedName(temp_name);
  const items = document.querySelectorAll('.picker-item');
  items.forEach((item) => {
    if (temp_name === item.textContent && temp_name != " · · · "){
      item.style.transform = 'scale(1.5)';
      item.style.backgroundColor = "turquoise";
    }
  });
  highlightAdditionalHoliday();
} //change function ends here

function scroll(info) {
  var result = '';
  // var isSpaceBeforeUppercase = false;

  for (var i = 0; i < info.length; i++) {
    var currentChar = info[i];
    var nextChar = info[i + 1];

    if (currentChar === ' ' && (nextChar.match(/[A-Z]/) )) {
      result += '&nbsp;&nbsp;&nbsp';
      // isSpaceBeforeUppercase = true;
    } else if (currentChar === ' '){
      result += '&nbsp';
    }else {
      result += currentChar;
      // isSpaceBeforeUppercase = false;
    }
  }
  var marquee = document.getElementById("scrollingText");
  marquee.innerHTML = '<p>' + result + '</p>';
}

const names = [
  " · · · ",
  " · · · ",
  "・・・",
  "詹文欽",
  "黃榮國",
  "范振宇",
  "唐＿茂",
  "許敦智",
  "王金誠",
  "王瑞發",
  "彭偉慎",
  "陳建中",
  "劉暐丞",
  "柯正和",
  "張日曜",
  "孫景泰",
  "秦桔萬",
  "邱冠霖",
  "林森發",
  "黃煜森",
  "劉錦郎",
  "余金原",
  "林厚運",
  "張哲維",
  "黃經洲",
  "洪柜峰",
  "林宏儒",
  "呂明峰",
  "周育稔",
  "許世勳",
  "羅應順",
  "方振彬",
  "・・・",
  "・・・",
];
const namePicker = document.getElementById("namePicker");
let currentIndex = 0;

const selectedClassName = 'selected';
const selectedClassName2 = 'selected2';

function highlightSelectedName(selectedName) {
  const days = document.querySelectorAll('.day');
  days.forEach(dayElement => {
    const date = `${year}-${month}-${dayElement.textContent}`;
    const scheduleForDay = dutySchedule[date] || '';

    // Split the schedule for the day into parts based on spaces
    const scheduleParts = scheduleForDay.split(' ');

    // Initialize two arrays to store names for each category
    const namesForSelectedClassName = [];
    const namesForSelectedClassName2 = [];

    // Initialize a flag to indicate whether to start categorizing into selectedClassName2
    let shouldCategorizeToClassName2 = false;

    // Loop through each part of the schedule
    scheduleParts.forEach(part => {
      // Check if the part starts with "S:" or "A:"
      if (part.startsWith('S:') || part.startsWith('A:')) {
        // Set the flag to true to start categorizing into selectedClassName2
        shouldCategorizeToClassName2 = true;
      } else if (shouldCategorizeToClassName2) {
        // If shouldCategorizeToClassName2 is true, categorize the part to selectedClassName2
        const name = part.trim();
        namesForSelectedClassName2.push(name);
        shouldCategorizeToClassName2 = false;
      } else {
        // If shouldCategorizeToClassName2 is false, categorize the part to selectedClassName
        const name = part.trim();
        namesForSelectedClassName.push(name);
      }
    });

    // Check if selectedName is in either category and apply appropriate class
    if (namesForSelectedClassName2.includes(selectedName)) {
      dayElement.classList.add(selectedClassName2);
    } else {
      dayElement.classList.remove(selectedClassName2);
    }

    if (namesForSelectedClassName.includes(selectedName)) {
      dayElement.classList.add(selectedClassName);
    } else {
      dayElement.classList.remove(selectedClassName);
    }

  });
}


function highlightAdditionalHoliday() {
  const days = document.querySelectorAll('.day');
  days.forEach(dayElement => {
    const date = `${year}-${month}-${dayElement.textContent}`;
    const namesForHoliday = (holiday[date] || '').split(' ');
    // console.log('namesForHoliday:', namesForHoliday);
    if (namesForHoliday.some(name => name.includes('放假日'))) {
      if (btn.checked) {
      dayElement.style.color = 'yellow';
      } else{
        dayElement.style.color = 'red';
      }
    }
    if (namesForHoliday.some(name => !name.includes('放假日'))) {
      if (btn.checked) {
        dayElement.style.color = 'white';
      } else {
        dayElement.style.color = 'black';
      }
    }

  });
}
// function AddLunar() {
//   const days = document.querySelectorAll('.day');
//   days.forEach(dayElement => {
//     const date = `${year}-${month}-${dayElement.textContent}`;
//     const lunarName = (holiday[date] || '').split(' ');
//     // console.log('namesForHoliday:', namesForHoliday);
//     dayElement.textContent += lunarName;
//   });
// }

// Populate the name picker with the list of names
names.forEach((name) => {
  const selectedName = name;
  const item = document.createElement("div");
  item.className = "picker-item";
  item.textContent = name;
  item.style.color = "gray";

 // Add click event listener to handle name selection
 item.addEventListener("click", () => {
    updateSelection();
    const items = document.querySelectorAll('.picker-item');
    items.forEach((item) => {
      item.style.transform = 'scale(1)';
      item.style.color = 'gray';
    });
    item.style.transform = 'scale(1.5)';
    item.style.color = '';
    item.style.backgroundColor = "turquoise";
    temp_name=selectedName;
```

```javascript
    highlightSelectedName(selectedName);
  });
  namePicker.appendChild(item);
});

function clearSelectedClass() {
  const days = document.querySelectorAll('.day');
  days.forEach(dayElement => {
    dayElement.classList.remove(selectedClassName);
  });
}

function updateSelection() {
  const items = document.querySelectorAll(".picker-item");
  items.forEach((item) => {
      item.style.backgroundColor = "";
  });
}


function updateScale() {
  const container = document.getElementById('namePicker');
  const items = document.querySelectorAll('.picker-item');
  const containerRect = container.getBoundingClientRect();
  const containerCenterY = containerRect.top + containerRect.height / 2;

  items.forEach((item) => {
    const itemRect = item.getBoundingClientRect();
    const itemCenterY = itemRect.top + itemRect.height / 2;
    const distanceToCenter = Math.abs(containerCenterY - itemCenterY);
      if (distanceToCenter < containerRect.height / 10 && item.textContent != "・・・"
      ) { // Adjust this threshold as needed
        item.style.transform = 'scale(1.5)';
        updateSelection();
        item.style.backgroundColor = "turquoise";
        item.style.color = '';
        temp_name=item.textContent;
        highlightSelectedName(item.textContent);
      } else {
        item.style.transform = 'scale(1)';
        item.style.color = 'gray';
      }
  });
}

namePicker.addEventListener("scroll", () => {
  const itemHeight = namePicker.querySelector(".picker-item").offsetHeight;
  currentIndex = Math.floor(namePicker.scrollTop / itemHeight);
  updateSelection();
  clearSelectedClass();
  updateScale();

});


const apiKey = '35af5c01f0d331eb99f5a42b0259c663';
const latitude = 25.07639;
const longitude = 121.22389;

// Clear: Indicates clear sky conditions.
// Clouds: Indicates cloudy weather.
// Rain: Indicates rainy weather.
// Drizzle: Indicates light rain.
// Thunderstorm: Indicates thunderstorm activity.
// Snow: Indicates snowy weather.
// Mist: Indicates misty or foggy conditions.
// Smoke: Indicates smoky conditions.
// Haze: Indicates hazy conditions.
// Dust: Indicates dusty or sandy conditions.
// Fog: Indicates foggy conditions.
// Sand: Indicates sandstorm conditions.
// Ash: Indicates volcanic ash in the air.
// Squall: Indicates sudden violent winds.
```

```javascript
//Tornado
// Function to fetch weather data from the API
function fetchWeather() {
  fetch(`https://api.openweathermap.org/data/2.5/weather?lat=${latitude}&lon=${longitude}&appid=${apiKey}`)
    .then(response => response.json())
    .then(data => {
       let temperatureKelvin = data.main.temp; // Temperature in Kelvin
       let temperatureCelsius = (temperatureKelvin - 273.15).toFixed(1);
       let ch_weather ='';
       let visibility = (data.visibility/1000).toFixed(1);
       let windSpeed = (data.wind.speed*1.943844).toFixed(2);; // Wind Speed in meters per second
      //  let windDirection = data.wind.deg; // Wind Direction in degrees

       humidity = data.main.humidity; // Humidity in percentage
       weatherCondition = data.weather[0].main;

      // Adjust background based on weather condition
      if (weatherCondition ==='Rain') {
        document.body.style.background = 'url(rain.png)';
        document.body.style.backgroundSize= 'cover';
        document.body.style.backgroundPosition= 'center';
        ch_weather='雨天';

      } else if (weatherCondition === 'Clouds' && humidity > 80) {
        document.body.style.background = 'url(clouds.png)';
        document.body.style.backgroundSize= 'cover';
        document.body.style.backgroundPosition= 'center';
        ch_weather='陰天';

      } else if (weatherCondition === 'Drizzle') {
        document.body.style.background = 'url(drizzle.png)';
        document.body.style.backgroundSize= 'cover';
        document.body.style.backgroundPosition= 'center';
        ch_weather='毛毛雨';

      } else if (weatherCondition === 'Thunderstorm') {
        document.body.style.background = 'url(thunderstorm.png)';
        document.body.style.backgroundSize= 'cover';
        document.body.style.backgroundPosition= 'center';
        ch_weather='雷雨';

      } else if (weatherCondition === 'Squall') {
        document.body.style.background = 'url(squall.png)';
        document.body.style.backgroundSize= 'cover';
        document.body.style.backgroundPosition= 'center';
        ch_weather='狂風暴雨';

      }  else {
        // Default background for other weather conditions
        document.body.style.background = 'url(tower.png)';
        document.body.style.backgroundSize= 'cover';
        document.body.style.backgroundPosition= 'center';
        ch_weather='晴天';
      }
      if(windSpeed > 63) {
        document.body.style.background = 'url(typhoon.png)';
        document.body.style.backgroundSize= 'cover';
        document.body.style.backgroundPosition= 'center';
        ch_weather='颱風天';

      }
      const date2 = `${year}-${month}-${day}`;
      let info = `${year}年${month}月${day}日`+ (holiday[date2] || '') + `桃園機場 ☛☛☛ 溫度:${temperatureCelsius}°C  濕度:${humidity}%   能見度:${visibility}km ${ch_weather}(${weatherCondition}) ☛☛☛`+ " " + (dutySchedule[date2] || '');
      scroll(info);
      // Log temperature and humidity

    })
    .catch(error => console.error('Error fetching weather:', error));
}


// Call fetchWeather function initially
fetchWeather();

// Call fetchWeather function periodically (e.g., every 10 minutes)
setInterval(fetchWeather, 600000); // 600000 milliseconds = 10 minutes


btn.addEventListener('change', change);
createCalendar(year, month);
// AddLunar();
highlightAdditionalHoliday(); //this must be done finally.
```