```
// Constants for the Seven Pi Over Four Calendar System
const DAYS_PER_WEEK = 7;
const DAYS_PER_MONTH = 30;
const MONTHS_PER_YEAR = 12;
const INTERCALARY_CYCLE = 7;
const INTERCALARY_MONTH_DAYS = 37;
const MICRO_YEAR_DAYS = 7;
const START_DATE = new Date(1913, 8, 15); // September 15, 1913 (Month is zero-based in
JavaScript)

function gregorianToSevenPiOverFour(gregorianDate) {
  const inputDate = new Date(gregorianDate);
  const deltaDays = Math.floor((inputDate - START_DATE) / (1000 * 60 * 60 * 24));

  // Determine how many full 7-year cycles have passed
  const sevenYearCycles = Math.floor(deltaDays / (INTERCALARY_CYCLE *
(MONTHS_PER_YEAR * DAYS_PER_MONTH) + INTERCALARY_MONTH_DAYS));

  // Calculate total intercalary days added
  const totalIntercalaryDays = sevenYearCycles * INTERCALARY_MONTH_DAYS;
  const adjustedDays = deltaDays + totalIntercalaryDays;

  // Find Seven Pi Over Four year
  const sevenPiYear = 1 + Math.floor(adjustedDays / (MONTHS_PER_YEAR *
DAYS_PER_MONTH + (sevenPiYear % INTERCALARY_CYCLE === 0 ?
INTERCALARY_MONTH_DAYS : 0)));
  const daysIntoYear = adjustedDays % (MONTHS_PER_YEAR * DAYS_PER_MONTH +
(sevenPiYear % INTERCALARY_CYCLE === 0 ? INTERCALARY_MONTH_DAYS : 0));

  // Determine if this is an intercalary year
  const isIntercalary = (sevenPiYear % INTERCALARY_CYCLE === 0);

  // Adjust for 13th Month in Intercalary Years
  let sevenPiMonth = Math.floor(daysIntoYear / DAYS_PER_MONTH) + 1;
  let sevenPiDay = (daysIntoYear % DAYS_PER_MONTH) + 1;

  if (isIntercalary && sevenPiMonth > 12) {
    sevenPiMonth = 13;
    sevenPiDay = daysIntoYear - (MONTHS_PER_YEAR * DAYS_PER_MONTH) + 1;
  }

  // Identify Micro Year (last 7 days of Month 13)
  let microYearDay = (isIntercalary && sevenPiMonth === 13 && sevenPiDay >
(INTERCALARY_MONTH_DAYS - MICRO_YEAR_DAYS)) ? sevenPiDay : null;
```

```
    // Determine week of macro year
    const weekOfMacroYear = Math.floor(adjustedDays / DAYS_PER_WEEK) + 1;

    // Determine day of the week (looping 1-7 continuously since start date)
    const dayOfWeek = (adjustedDays % DAYS_PER_WEEK) + 1;

    return {
        year: sevenPiYear,
        month: sevenPiMonth,
        day: sevenPiDay,
        week_of_macro_year: weekOfMacroYear,
        day_of_week: dayOfWeek,
        intercalary_year: isIntercalary,
        micro_year_day: microYearDay
    };
}

function updateCurrentDate() {
    const today = new Date();
    const convertedDate = gregorianToSevenPiOverFour(today);
    document.getElementById("currentDate").textContent =
        `Year: ${convertedDate.year}, Month: ${convertedDate.month}, Day: ${convertedDate.day},
         Week: ${convertedDate.week_of_macro_year}, Day of Week:
${convertedDate.day_of_week}`;
}

function convertDate() {
    const inputDate = document.getElementById("gregorianDate").value;
    if (inputDate) {
        const convertedDate = gregorianToSevenPiOverFour(inputDate);
        let resultText = `Year: ${convertedDate.year}, Month: ${convertedDate.month}, Day:
${convertedDate.day},
                          Week: ${convertedDate.week_of_macro_year}, Day of Week:
${convertedDate.day_of_week}`;

        if (convertedDate.intercalary_year) {
            resultText += ` (Intercalary Year)`;
            if (convertedDate.micro_year_day !== null) {
                resultText += ` | Micro Year Day: ${convertedDate.micro_year_day}`;
            }
        }

        document.getElementById("conversionResult").textContent = resultText;
    } else {
        document.getElementById("conversionResult").textContent = "Please enter a valid date.";
    }
}

updateCurrentDate();
```