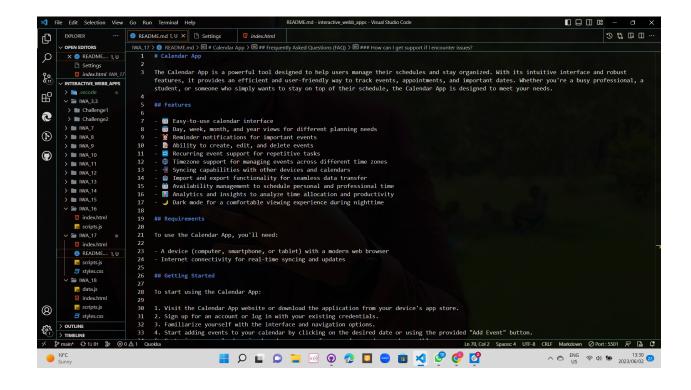# DWA_03.4 Knowledge Check_DWA3.1

_____

1. Please show how you applied a Markdown File to a piece of your code.

_____

2. Please show how you applied JSDoc Comments to a piece of your code.

```javascript
 76        }
 77        return weeks;
 78    };
 79
 80    /**
 81     * Add a table cell to an existing string.
 82     *
 83     * @param {string} existing - The existing string.
 84     * @param {string} classString - The CSS class string for the cell.
 85     * @param {string|number} value - The value of the cell.
 86     * @returns {string} The updated string with the added table cell.
 87     */
 88    const addCell = function(existing, classString, value) {
 89        return `${existing}<td class="${classString}">${value}</td>`;
 90    };
 91
 92    /**
 93     * Create HTML table string from the data.
 94     *
 95     * @param {Object[]} data - The array of week objects containing day information.
 96     * @returns {string} The HTML table string.
 97     */
 98    const createHtml = function(data) {
 99        let result = '';
100        for (let i = 0; i < data.length; i++) {
101            const week = data[i];
102            let inner = '';
103            inner = addCell(inner, 'table__cell table__cell_sidebar', `Week ${week.week}`);
104            for (let j = 0; j < week.days.length; j++) {
105                const day = week.days[j];
106                let classString = 'table__cell';
107                const currentDate = new Date();
108                const isToday = currentDate.getDate() === day.value && currentDate.getMonth() === currentDate.getMonth();
109                const isWeekend = day.dayOfWeek === 1 || day.dayOfWeek === 7;
110                const isAlternate = week.week % 2 === 0;
```
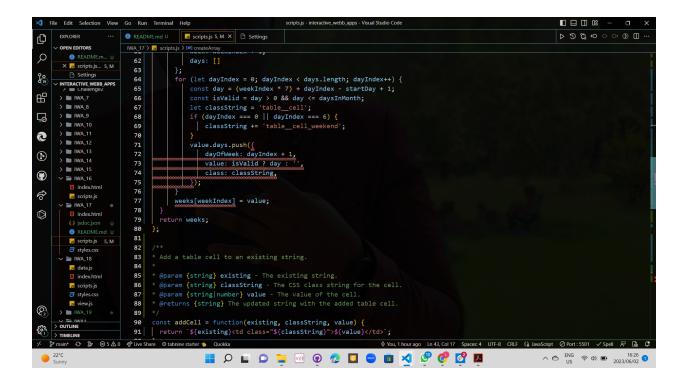
3. Please show how you applied the @ts-check annotation to a piece of your code.

```javascript
          days: []
      };
      for (let dayIndex = 0; dayIndex < days.length; dayIndex++) {
          const day = (weekIndex * 7) + dayIndex - startDay + 1;
          const isValid = day > 0 && day <= daysInMonth;
          let classString = 'table__cell';
          if (dayIndex === 0 || dayIndex === 6) {
              classString += 'table__cell_weekend';
          }
          value.days.push({
              dayOfWeek: dayIndex + 1,
              value: isValid ? day : '',
              class: classString,
          });
      }
      weeks[weekIndex] = value;
  }
  return weeks;
};

/**
 * Add a table cell to an existing string.
 *
 * @param {string} existing - The existing string.
 * @param {string} classString - The CSS class string for the cell.
 * @param {string|number} value - The value of the cell.
 * @returns {string} The updated string with the added table cell.
 */
const addCell = function(existing, classString, value) {
  return `${existing}<td class="${classString}">${value}</td>`;
```

_____

4. As a BONUS, please show how you applied any other concept covered in the 'Documentation' module.

Features

- 📅 Easy-to-use calendar interface
- 📆 Day, week, month, and year views for different planning needs
- ⏰ Reminder notifications for important events
- 📝 Ability to create, edit, and delete events
- 🔁 Recurring event support for repetitive tasks
- 🌐 Timezone support for managing events across different time zones
- 🔗 Syncing capabilities with other devices and calendars
- 📤 Import and export functionality for seamless data transfer
- 📅 Availability management to schedule personal and professional time
- 📊 Analytics and insights to analyze time allocation and productivity
- 🌙 Dark mode for a comfortable viewing experience during nighttime

Requirements

To use the Calendar App, you'll need:

- A device (computer, smartphone, or tablet) with a modern web browser
- Internet connectivity for real-time syncing and updates

Getting Started

To start using the Calendar App:

1. Visit the Calendar App website or download the application from your device's app store.
2. Sign up for an account or log in with your existing credentials.