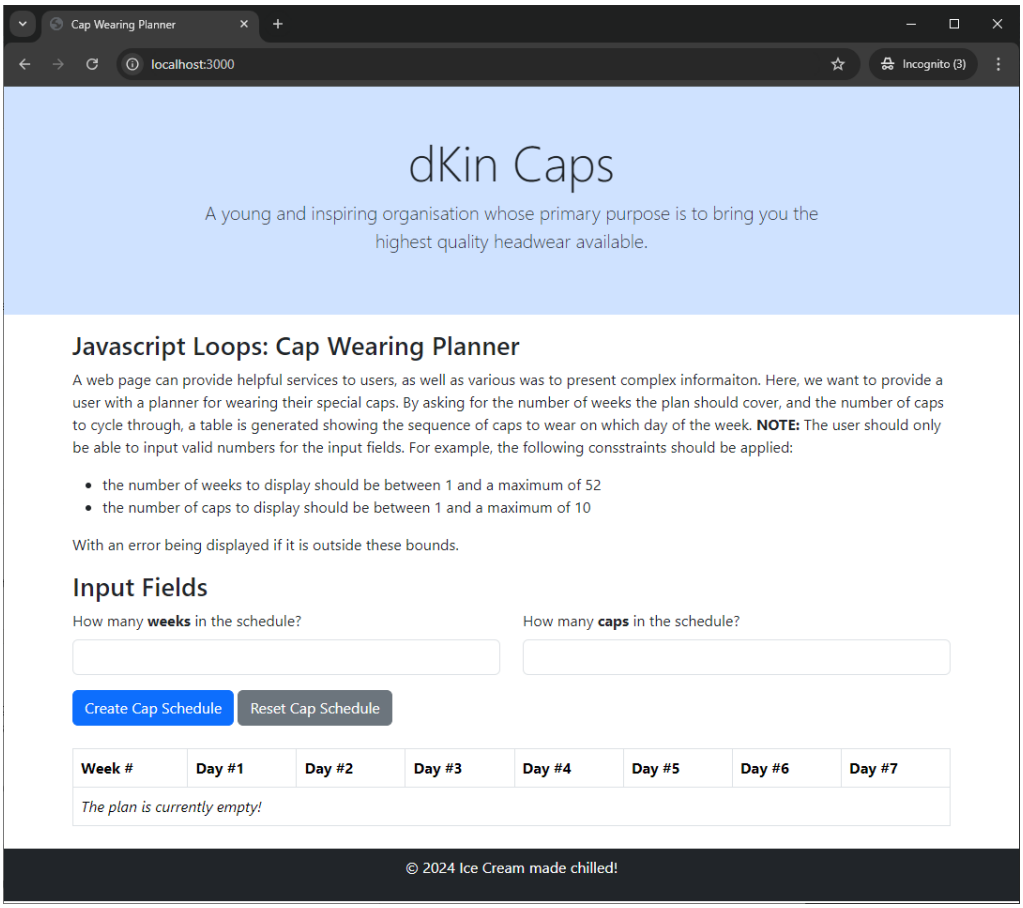
# 6.2C: JavaScript Loop

## Tasks

In this task you are asked to construct a **Cap Wearing Planner** in the form of a table, that is dynamically created based on (valid) input from the user entered through a simple form.

You are provided with a starting web page, that contains a form (two fields) and a list that is already populated with one rating. The sample starting page should look like this:

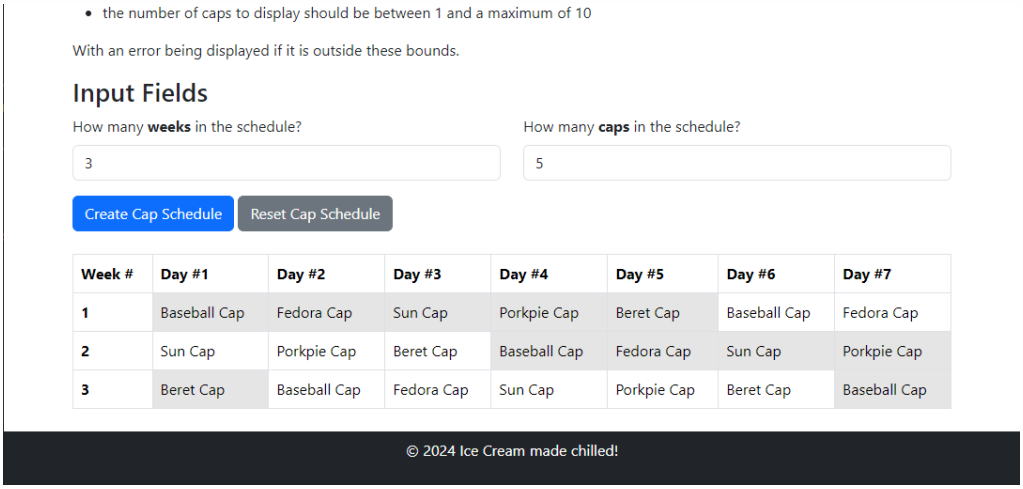



Task6.2.1: Initial page view

The form accepts from the user two number values, the `Number of weeks` and the `Number of caps`, which relate to how long the planner should be, and the number of caps to cycle through, respectively. With this information a table with 8-columns is created that shows:

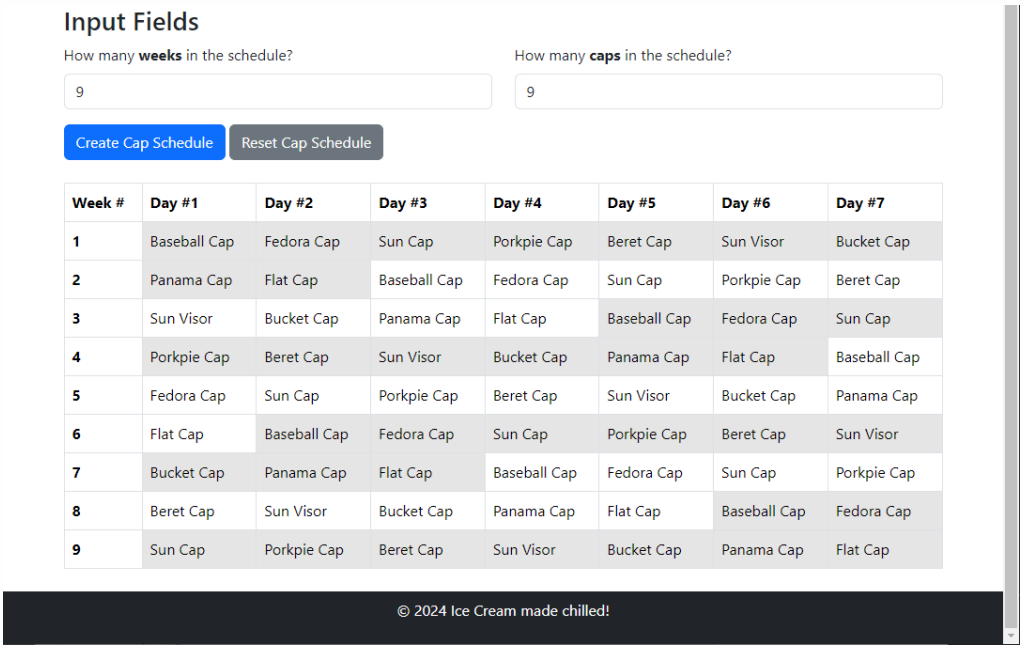
1. The week number
2. The type of cap to wear on that day of the week (cycling through the number of caps owned: maximum 10)
3. Highlighting the alternate cap sequences with those cells having a grey (`table-active`) and white backgrounds.

The following screenshots show the result of three example inputs. The first is a *3 week plan, with 5 caps*:



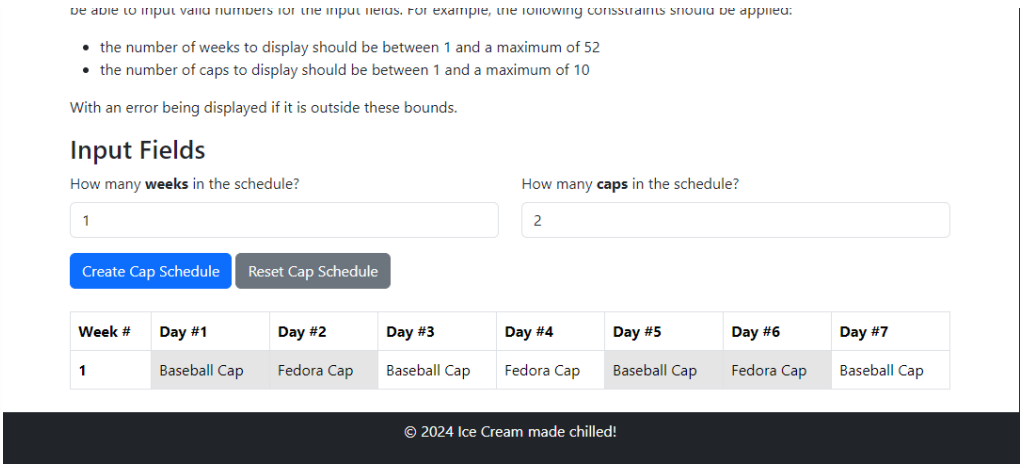

Task6.2.2: Input of 3 week plan, with 5 caps

The next is a *9 week plan, with 9 caps*:




Task6.2.3 Input of 9 week plan, with 9 caps

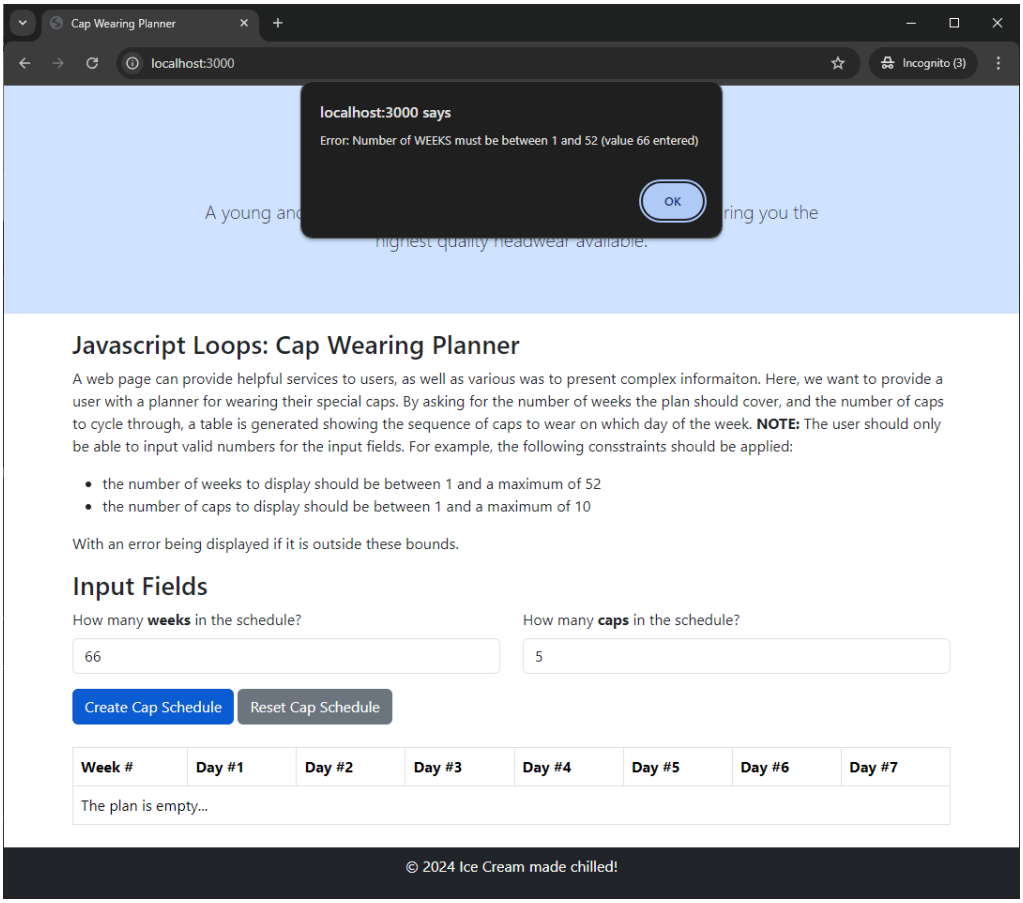
And the final screenshot shows a *1 week plan, with 2 caps*:




Task6.2.4 Input of 1 week plan, with 2 caps

In completing this task, the code used to dynamically create (extend) the table should be allocated within a javascript function. This function is called when the user clicks the `Create Cap Schedule` button.

In adding to the list, the javascript code should take and process the input fields from the form (`weekCountInput` and `capCountInput`). The *week count* input must be check to be within the 1..52 range and if it is outside this range an error message provided. Similarly, the *cap count* input must be check to be within the 1..10 range and if it is outside this range an error message provided. In the case of an error case for either value, return the user to the inut form (`return` in the javascript function). A sample of the error prompt provided could be:




Task6.2.5 Error message on invalid *rating* value

## Hints

In completing this task consider the following points:

- Include the array of cap names from Task 6.1P (list of 10 caps)
- Build your code in stages, i.e., get and check input from the user; find the fields within the HTML document that need to be updated (e.g., the ID `cap-planner-table-body`)
- The `input` fields of the form can be extracted using DOM commands such as
  - `let weekCountMax = parseInt( document.getElementById("weekCountInput").value );`
  - `let capCountMax = parseInt( document.getElementById("capCountInput").value );`
- The `weekCountMax` value inputted by the user should be checked to see it is **invalid** (i.e., `!weekCountMax` or `< 1` or `> 52`) and if it is an error message should be provided (using `window.alert()`) then returning from the function.
- The `capCountMax` value inputted by the user should be checked to see it is **invalid** (i.e., `!capCountMax` or `< 1` or `> 10`) and if it is an error message should be provided (using `window.alert()`) then returning from the function.
- It may be useful to have nested double loop to construct your table. Outer loop creates a row for each of the number of weeks, while the inner creates a table cell for each the 7 days.
- Keep track of the *cell count* and use the *modulus* operator to see if cycles of `capCountMax` have been reached before toggling between *greyed* and *white* backgrounds.
- Online references include:
  - (https://www.w3schools.com/jsref/dom_obj_table.asp)
  - (https://www.w3schools.com/js/js_arithmetic.asp) Modulus (remainder) operator

# What will you submit?

You should submit:

- Screen-shot of the web page showing the output for 5 different cap planner inputs.
- Screen-shot of the page when an 'invalid' number of weeks is entered (i.e., a week count of `-1` or `99`) and the prompt warning window is presented.
- The webpage's `HTML` file.
- The file with `JavaScript` code used to build the table.