

- Feature Implementation: Dynamic Trip Allocation Calendar
 - Overview
 - Technology Stack
 - Backend Logic
 - Endpoint: `/schedule/tripsallocated/range`
 - Method: `POST`
 - Frontend Logic
 - Initialization of Flatpickr
 - Function: `displayCalendarGridRange`
 - Problem Faced
 - Error: `Uncaught ReferenceError: startDate is not defined`
 - Additional Concern
 - Outcome

Feature Implementation: Dynamic Trip Allocation Calendar

Overview

This document explains the implementation of a feature that integrates a Flatpickr date picker with dynamic backend data to show available trip allocations per day. The goal was to enhance user experience by allowing selection of valid dates only, showing availability tooltips, and styling based on trip count.

Technology Stack

- **Frontend:** JavaScript, jQuery, Flatpickr
- **Backend:** Laravel (PHP)

Backend Logic

Endpoint:

/schedule/tripsallocated/range

Method: POST

```
public function trips_allocated_range (Request $request)
{
    $start_date = $request->start_date;
    $end_date = $request->end_date;

    $current_date = $start_date;
    $trips_allocated = [];
    while ($current_date){
        $remaining_trips = $this->trips_allocated($current_date);
        $trips_allocated[$current_date] = $remaining_trips;
        $current_date = Carbon::createFromFormat('Y-m-d', $current_date)->addDay()->format('Y-m-d');
        if ($current_date > $end_date) {
            break;
        }
    }
    return response()->json($trips_allocated);
}
```

Frontend Logic

Initialization of Flatpickr

```
flatpickr('.flatpickr-reschedule', {
    dateFormat: 'Y-m-d',
    minDate: 'today',
    allowInput: true,
    onReady: function (selectedDates, dateStr, instance) {
        if (instance.input.id === 'proposed_emptying_date') {
            displayCalendarGridRange(instance);
        }
    },
    onMonthChange: function (selectedDates, dateStr, instance) {
        if (instance.input.id === 'proposed_emptying_date') {
            displayCalendarGridRange(instance);
        }
    }
});
```

```
    }  
  });  
}
```

Function:

`displayCalendarGridRange`

```
function displayCalendarGridRange(instance) {  
  const calendarContainer = instance.calendarContainer;  
  const dayElements = calendarContainer.querySelectorAll(".flatpickr-day");  
  if (dayElements.length === 0) return;  
  
  const firstVisibleDay = dayElements[0].dateObj;  
  const lastVisibleDay = dayElements[dayElements.length - 1].dateObj;  
  
  const formatYMD = (d) => `${d.getFullYear()}-${String(d.getMonth() + 1).padStart(2, '0')}-${String(d.getDate()).padStart(2, '0')}`;  
  
  const startDateFormattedYMD = formatYMD(firstVisibleDay);  
  const endDateFormattedYMD = formatYMD(lastVisibleDay);  
  
  $.ajax({  
    url: "{{ route('schedule.tripsallocated.range') }}",  
    type: 'POST',  
    data: {  
      start_date: startDateFormattedYMD,  
      end_date: endDateFormattedYMD  
    },  
    headers: {  
      'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')  
    },  
    dataType: 'json',  
    success: function(response) {  
      const tripData = response;  
      const disabledDates = Object.keys(tripData).filter(date =>  
tripData[date] === 0);  
  
      flatpickr("#proposed_emptying_date", {  
        dateFormat: "Y-m-d",  
        allowInput: true,  
        minDate: startDateFormattedYMD,  
        maxDate: endDateFormattedYMD,  
        disable: disabledDates,  
        onDayCreate: function (dObj, dStr, fp, dayElem) {  
          const date = dayElem.dateObj.toISOString().slice(0, 10);  
          if (tripData[date] !== undefined) {  
            const trips = tripData[date];  
            dayElem.setAttribute("title", `Trips Available: ${trips}`);  
  
            if (trips === 0) {  
              dayElem.style.backgroundColor = "#f8d7da";  
              dayElem.style.color = "#721c24";  
            }  
          }  
        }  
      });  
    }  
  });  
}
```

```
        } else if (trips <= 2) {
            dayElem.style.backgroundColor = "#fff3cd";
            dayElem.style.color = "#856404";
        } else {
            dayElem.style.backgroundColor = "#d4edda";
            dayElem.style.color = "#155724";
        }
        dayElem.style.borderRadius = "50%";
    }
    });
}
```

Problem Faced

Error: Uncaught ReferenceError: startDate is not defined

- **Reason:** The variable startDate was referenced instead of the correctly scoped startDateFormattedYMD.
- **Fix:** Replaced undefined variable with the correct one.

Additional Concern

- Avoided **Flatpickr reinitialization** inside the AJAX success callback to prevent flickering and event duplication.

Outcome

- Calendar now shows disabled dates based on backend trip availability.
- Users get hover tooltips showing available trips.
- Calendar dates are visually styled to indicate low, medium, or no availability.

Let me know if you'd like to include screenshots or want this tailored for #supervisory_assessment_date too.