



How to Use React Day Picker to Disable Dates After Adding Events



Goal

Create a calendar where:

- The user can **click a date to add an event**.
- Once an event is added to that date, the day becomes **grayed out and disabled**.
- The user **cannot add more than one event per day**.



Step-by-Step Implementation

1. Install React Day Picker

If you haven't already, install the dependency:

Shell

```
npm install react-day-picker
```

2. Set Up the Calendar Component

Start by importing and displaying the `<DayPicker />` component:

None

```
import { useState } from 'react';
import { DayPicker } from 'react-day-picker';
import 'react-day-picker/dist/style.css'; // default styles

function EventCalendar() {
  const [bookedDates, setBookedDates] = useState([]);

  function handleDayClick(day, modifiers) {
    if (modifiers.disabled) {
```

```

        return; // Prevent selecting disabled days
    }

    // Add the event here – could open a modal or simple prompt
    const title = prompt("Enter event title:");
    if (title) {
        setBookedDates([...bookedDates, day]);
        // Optional: store the title if you want to show it later
    }
}

return (
    <div>
        <h2>Click a day to add an event</h2>
        <DayPicker
            onClick={handleDayClick}
            disabled={bookedDates}
            modifiersClassNames={{ disabled: 'grayed-out' }}
        />
    </div>
);
}

```

3. Add Custom CSS to Gray Out Disabled Days

You can customize the appearance of disabled days with a simple CSS class:

```

CSS
.grayed-out {
    color: #bbb;
    background-color: #f5f5f5;
    text-decoration: line-through;
    pointer-events: none;
}

```

Make sure this CSS is included in your component or global stylesheet.

How It Works

- **bookedDates** is a list of all dates that already have events.
 - The **disabled** prop on `<DayPicker />` uses this list to disable (gray out and block) those dates.
 - The **onDayClick** function checks if the clicked day is disabled:
 - If it's not, it prompts the user to add an event.
 - Once added, the date is added to **bookedDates**, disabling it for the future.
-

Optional Add-ons

- Store event **titles** along with the dates (e.g., as `{ date: day, title: 'Meeting' }`) if you want to display them.
 - Use a modal instead of **prompt()** to make it more user-friendly.
 - Save events to a database or `localStorage` to persist them across page reloads.
-

Summary

Feature	Implementation
Display calendar	<code><DayPicker /></code> from <code>react-day-picker</code>
Disable selected days	<code>disabled={bookedDates}</code>
Gray out disabled days	CSS with <code>modifiersClassNames</code>
Add events on click	<code>onDayClick</code> handler with a check for <code>disabled</code>