

```
npm install @fullcalendar/react @fullcalendar/daygrid @fullcalendar/timegrid
@fullcalendar/interaction
npm install @mui/material @emotion/react @emotion/styled
```

```
// CalendarPlanner.tsx
import React, { useState } from 'react';
import FullCalendar, { DateSelectArg, EventClickArg, EventInput } from '@fullcalendar/react';
import dayGridPlugin from '@fullcalendar/daygrid';
import timeGridPlugin from '@fullcalendar/timegrid';
import interactionPlugin from '@fullcalendar/interaction';
import { Box, Dialog, DialogTitle, DialogContent, TextField, DialogActions, Button } from
 '@mui/material';
```

```
const CalendarPlanner: React.FC = () => {
  const [events, setEvents] = useState<EventInput[]>([]);
  const [open, setOpen] = useState(false);
  const [selectedDate, setSelectedDate] = useState<DateSelectArg | null>(null);
  const [title, setTitle] = useState("");
```

```
  const handleDateSelect = (selectInfo: DateSelectArg) => {
    setSelectedDate(selectInfo);
    setTitle("");
    setOpen(true);
  };
```

```
  const handleEventClick = (clickInfo: EventClickArg) => {
    if (window.confirm(`Delete event '${clickInfo.event.title}'?`)) {
      clickInfo.event.remove();
    }
  };
```

```
  const handleAddEvent = () => {
    if (selectedDate && title.trim()) {
      const calendarApi = selectedDate.view.calendar;
      calendarApi.unselect();
```

```
      calendarApi.addEvent({
        id: String(new Date().getTime()),
        title,
        start: selectedDate.startStr,
        end: selectedDate.endStr,
        allDay: selectedDate.allDay,
      });
```

```
      setOpen(false);
    }
  };
};
```

```

return (
  <Box sx={{ p: 2 }}>
    <FullCalendar
      plugins={[dayGridPlugin, timeGridPlugin, interactionPlugin]}
      initialView="dayGridMonth"
      selectable={true}
      editable={true}
      events={events}
      select={handleDateSelect}
      eventClick={handleEventClick}
      height="auto"
    />

    <Dialog open={open} onClose={() => setOpen(false)}>
      <DialogTitle>Add Event</DialogTitle>
      <DialogContent>
        <TextField
          autoFocus
          margin="dense"
          label="Event Title"
          type="text"
          fullWidth
          value={title}
          onChange={(e) => setTitle(e.target.value)}
        />
      </DialogContent>
      <DialogActions>
        <Button onClick={() => setOpen(false)}>Cancel</Button>
        <Button onClick={handleAddEvent} variant="contained">Add</Button>
      </DialogActions>
    </Dialog>
  </Box>
);
};

```

```

export default CalendarPlanner;

```