<https://chatgpt.com/c/683aae49-ee64-800d-a106-4c91b6f4fa84>

why its not showing with other required field showing when they empty.. it is only showing once i filled all other requried fields like doctor patinet etc and submit it with blank date

**ChatGPT said:**

Great question — the reason your **custom appointment date validation message** is not showing alongside other required fields when the user first clicks “Submit” is because **React Hook Form only triggers its built-in validation** for fields it knows about — and your custom date picker (CustomDatePicker) is not registered using react-hook-form.

So when the user hits "Submit", react-hook-form checks its registered fields (like doctor and patient), and only *after* they pass, your onSubmit function is even called — and **only then** does your custom date validation run.

**✅ Solution Options**

**Option 1: Integrate CustomDatePicker using Controller**

You can wrap your date picker in a Controller provided by react-hook-form. This way, the form will treat the date picker like a native field — and validate it properly with the others.

Here’s how to do it:

**🔧 Modified Code (Using Controller)**

1. **Import Controller from react-hook-form:**

js

CopyEdit

import { useForm, Controller } from "react-hook-form";

1. **Replace your date picker JSX with this:**

jsx

CopyEdit

<Form.Group className="mb-3">

<Form.Label>Appointment Date</Form.Label>

<Controller

name="appointmentDate"

control={control}

rules={{ required: "Appointment date is required" }}

render={({ field }) => (

<>

<CustomDatePicker

dob={field.value}

setDob={(val) => field.onChange(val)}

disabled={{ after: new Date() }}

/>

{errors.appointmentDate && (

<div className="text-danger" style={{ fontSize: "0.875em" }}>

{errors.appointmentDate.message}

</div>

)}

</>

)}

/>

</Form.Group>

1. **Update onSubmit** to use the appointmentDate from data directly:

js

CopyEdit

const onSubmit = (data) => {

const formattedDate = format(data.appointmentDate, "yyyy-MM-dd");

if (appointment) data.id = appointment.id;

onSave({

...data,

appointmentDate: formattedDate,

doctorId: parseInt(data.doctorId),

patientId: parseInt(data.patientId),

});

};

1. **Initialize default value for appointmentDate:**

Update your form initialization (optional):

js

CopyEdit

const {

register,

handleSubmit,

setValue,

control,

formState: { errors },

} = useForm({

defaultValues: {

appointmentDate: null,

},

});

**✅ Result**

Now, the date field will behave **just like all other required fields** — its validation message will appear immediately when the user submits an empty form.

Let me know if you’d like me to help refactor CustomDatePicker slightly to work even more smoothly with this controller approach.

**You said:**

go ahead

**ChatGPT said:**

Great — let’s adjust your CustomDatePicker component so it works cleanly with react-hook-form’s Controller. Here's what we'll do:

**✅ Goals for CustomDatePicker**

* Accept value and onChange props (just like a normal input)
* Be reusable in controlled forms
* No need to manage dob/setDob outside — it will be fully controlled by the form

**🔧 Updated CustomDatePicker.jsx**

jsx

CopyEdit

// File: src/components/Shared/CustomDatePicker.jsx

import React from "react";

import { DayPicker } from "react-day-picker";

import "react-day-picker/dist/style.css";

const CustomDatePicker = ({ value, onChange, disabled }) => {

const handleDaySelect = (date) => {

onChange(date);

};

return (

<DayPicker

mode="single"

selected={value}

onSelect={handleDaySelect}

disabled={disabled}

/>

);

};

export default CustomDatePicker;

**✅ Then, in AppointmentForm.jsx, use it like this:**

jsx

CopyEdit

<Controller

name="appointmentDate"

control={control}

rules={{ required: "Appointment date is required" }}

render={({ field }) => (

<>

<CustomDatePicker

value={field.value}

onChange={field.onChange}

disabled={{ after: new Date() }}

/>

{errors.appointmentDate && (

<div className="text-danger" style={{ fontSize: "0.875em" }}>

{errors.appointmentDate.message}

</div>

)}

</>

)}

/>

**🔄 Final Cleanup**

You can now **remove** the dob and setDob state from AppointmentForm.jsx, since react-hook-form is handling the state for you.

Also, in your onSubmit, just format:

js

CopyEdit

const formattedDate = format(data.appointmentDate, "yyyy-MM-dd");

Let me know if you'd like me to post a full cleaned-up version of your AppointmentForm.jsx after this refactor.