**Roteiro da Aula 4**

**FRONTEND**

1. npm install @radix-ui/react-checkbox
2. alterar arquivo HabitDay.tsx, depois do ProgressBar

<div className="mt-6 flex flex-col gap-3">

<Checkbox.Root className="flex items-center gap-3 group">

<div

className="h-8 w-8 rounded-lg flex items-center justify-center bg-zinc-900 border-2 border-zinc-800 group-data-[state=checked]:bg-green-500 group-data-[state=checked]:border-green-500">

<Checkbox.Indicator>

<Check size={20} className="text-white"/>

</Checkbox.Indicator>

</div>

<span className="font-semibold text-xl text-white leading-tight group-data-[state=checked]:line-through group-data-[state=checked]:text-zinc-400">

Beber 2L de água

</span>

</Checkbox.Root>

</div>

1. Alterar arquivo NewHabitForm.tsx, criando a variável

const availableWeekDays = [

'Domingo',

'Segunda-feira',

'Terça-feira',

'Quarta-feira',

'Quinta-feira',

'Sexta-feira',

'Sábado'

]

1. Alterar arquivo NewHabitForm.tsx, abaixo de Qual a recorrência?

<div className="flex flex-col gap-2 mt-3">

{availableWeekDays.map((weekDay, index) => {

return (

<Checkbox.Root

key={weekDay}

className="flex items-center gap-3 group"

checked={weekDays.includes(index)}

onCheckedChange={() => handleToggleWeekDay(index)}

>

<div className="h-8 w-8 rounded-lg flex items-center justify-center bg-zinc-900 border-2 border-zinc-800 group-data-[state=checked]:bg-green-500 group-data-[state=checked]:border-green-500">

<Checkbox.Indicator>

<Check size={20} className="text-white"/>

</Checkbox.Indicator>

</div>

<span className="text-white leading-tight">

{weekDay}

</span>

</Checkbox.Root>

)

})}

</div>

1. Alterar arquivo NewHabitForm.tsx

const [title, setTitle] = useState('')

const [weekDays, setWeekDays] = useState<number[]>([])

1. Alterar o arquivo NewHabitForm.tsx, alterando o input:

<input

type="text"

id="title"

placeholder="ex.. Exercícios, dormir bem, etc..."

className="p-4 rounded-lg mt-3 bg-zinc-800 text-white placeholder:text-zinc-400"

autoFocus

value={title}

onChange={event => setTitle(event.target.value)}

/>

1. Alterar o arquivo NewHabitForm.tsx, alterando o checkbox:

<Checkbox.Root

key={weekDay}

className="flex items-center gap-3 group"

checked={weekDays.includes(index)}

onCheckedChange={() => handleToggleWeekDay(index)}

>

8. e adicionando a função:

function handleToggleWeekDay(weekDay: number) {

if (weekDays.includes(weekDay)) {

const weekDaysWithRemovedOne = weekDays.filter(day => day !== weekDay)

setWeekDays(weekDaysWithRemovedOne)

} else {

const weekDaysWithAddedOne = [...weekDays, weekDay]

setWeekDays(weekDaysWithAddedOne)

}

}

9. npm install axios

10. na pasta lib/axios.ts

import axios from 'axios'

export const api = axios.create({

baseURL: 'http://localhost:3333'

})

11. Alterar o arquivo SummaryTable.tsx

import { api } from '../lib/axios'

type Summary = Array<{

id: string

date:string

amount: number

completed: number

}>

export function SummaryTable() {

const [summary, setSummary] = useState<Summary>([])

useEffect(() => {

api

.get('summary')

.then(response => {

setSummary(response.data)

} )

}, [])

...

<div className="grid grid-rows-7 grid-flow-col gap-3">

{summaryDates.map(date => {

const dayInSummary = summary.find(day => {

return dayjs(date).isSame(day.date, 'day')

})

return (

<HabitDay

key={date.toString()}

date={date}

amount={dayInSummary?.amount}

completed={dayInSummary?.completed}

/>

)

})}

12. alterar HabitDayProps:

interface HabitDayProps {

date: Date

completed?: number

amount?: number

}

export function HabitDay({ completed = 0, amount = 0, date }: HabitDayProps) {

const completedPercentage = amount > 0 ? Math.round((completed / amount) \* 100) : 0

const dayAndMonth = dayjs(date).format('DD/MM')

const dayOfWeek = dayjs(date).format('dddd')

<span className="font-semibold text-zinc-400">{dayOfWeek}</span>

<span className="mt-1 font-extrabold leading-tight text-3xl">{dayAndMonth}</span>

13. criar arquivo lib/dayjs.ts

import dayjs from "dayjs"

import 'dayjs/locale/pt-br'

dayjs.locale('pt-br')

14. Alterar arquivo App.tsx

import './lib/dayjs'

15. Alterar arquivo NewHabitForm.tsx

async function createNewHabit(event: FormEvent) {

event.preventDefault()

if (!title || weekDays.length === 0) {

return

}

await api.post('habits', {

title,

weekDays

})

setTitle('')

setWeekDays([])

alert('Hábito criado com sucesso!')

}