Challenge: Calendar Kanban Board!

Summary: You are tasked with creating a calendar with draggable events and an expanded event detail screen.

Requirements:

1. Build a kanban board calendar experience that is safe for web and mobile
    a. On mobile, show a single day
        i. If you swipe left or right, you should be able to go to the next or previous day. Watch out for how it should work when you start your touch event on the card - since they are also draggable elements.
    b. On desktop, show a full calendar week
        i. Can go to next or previous week using
2. You should be able to drag the cards to the left or the right to view other days and easily place them on the other days - both on mobile & desktop.
    a. By bringing the card to the left or right shoulder of the screen, you should be able to “traverse” to the next day if you hold there for 1.5s or whatever timing feels natural to you while prototyping (Google Calendar has a great example for this on their mobile app)
3. When clicking to open up an event, you should transition that specific card to a full screen view of the same card, with a smooth card-to-detail transition.
    a. Essentially, there should be no “sudden” changes, like a distinct full page change. it should nicely animate from the card you clicked to the full screen view of that card and its details.
    b. Bonus: The items on the card can also nicely animate to wherever they need to be in the detailed view. You can make calls on how this should look best or what would be the best visual experience based on your own trial & error, and to your taste.
4. The dragging should be smooth across desktop or mobile
5. Nothing needs to be persisted or saved, other than some minimal local state, you can hardcode the initial values, and no backend or auth requirements at all.
6. There should be a top header which shows the full week (shown in Mock UI below) - and as you go from day to day, this should also animate based on the state (ie, the current day).
7. Reordering in a given day is not a requirement. The time field should be used for the order.

Success Criteria:

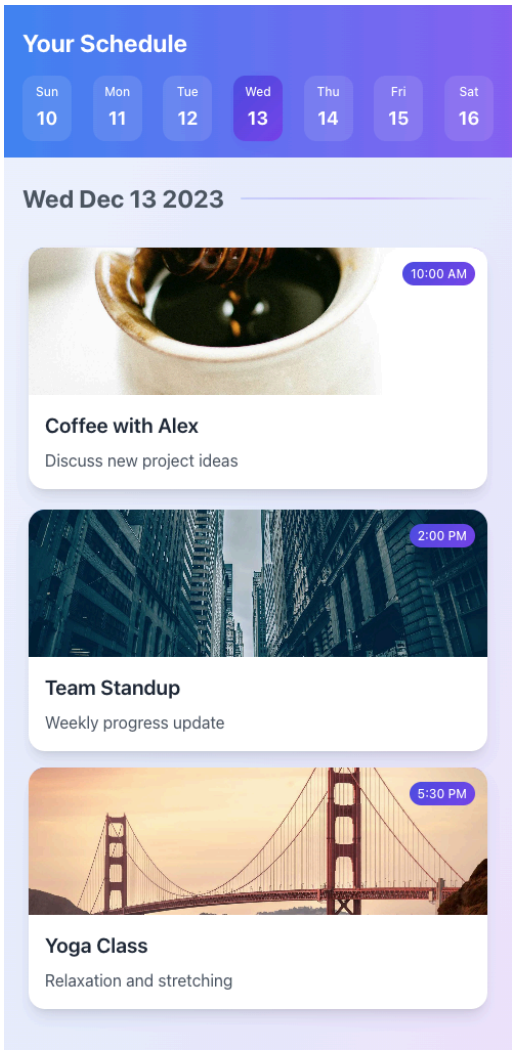
- Is this something you’d launch to users and be proud of?
- Does the user experience work really well and smoothly? Is it bug-free across platforms?
- Is the code easy to reason about, well encapsulated, and respect separation of concerns between components / shared state logic, etc?

- We don’t care too much about design / looks as that’s not what you are being tested on - as long as it looks presentable… the more important piece to focus on is your craftsmanship when it comes to building performant, reliable, robust user interfaces.
- Add animations and other frontend niceties where it makes sense! We’re especially excited to see how you do the card-to-detail transition, the transitions as you switch days, and the dragging between days logic…
    - The dragging features are the core thing to focus on.
- Infer what you can - we’ve given you a mock for mobile, use your imagination for desktop!
- If you run into any roadblocks, or if you feel that certain elements are impossible - describe the challenges to the leadership team and propose another direction you feel is more suitable if that is the case.

Things to avoid:

- Rabbit holes. There is ambiguity in this challenge to test your taste & knowing where to look, but also tests whether you can know when to not go down a rabbit hole.
- Don’t be afraid to ask questions! Clarify things… pretend like we are working together on this…

An example of the UI could look like this, but feel free to change it to your taste, or if you think it would work better another way:



The desktop version can look like this but with a 7-day view instead, and a back & forward arrow to go through weeks instead of any swipe behaviour.

Here is an example of what a card-to-detail transition could look like once you click on the card:

- https://dribbble.com/shots/4564728-Fluid-Parallax-Card-Transition-Flow

You can add more data if you'd like it to appear more full in the detailed view, just ask Cursor / ChatGPT to add mock data you'd like.

Here is the mock data we are providing you with:

```
interface Event {
  id: string;
  title: string;
  description: string;
  imageUrl: string;
  time: string;
}

interface EventsByDate {
  [date: string]: Event[];
}

const events: EventsByDate = {
  "2024-03-11": [
    {
      id: "event-1",
      title: "Coffee with Alex",
      description:
        "Meet with Alex to brainstorm ideas for the upcoming product
launch. We'll review market research and competitor analysis to identify
potential opportunities and challenges.",
      imageUrl:

"https://fastly.picsum.photos/id/312/1920/1080.jpg?hmac=OD_fP9MUQN7uJ8NBR7t
lii78qwHPUROGgohG4w16Kjw",
      time: "09:00 AM",
    },
    {
      id: "event-2",
      title: "Team Standup",
      description:
        "Weekly standup meeting with the dev team. Discuss progress,
```

```
blockers, and align on next week's priorities.",
      imageUrl:

"http://fastly.picsum.photos/id/737/1920/1080.jpg?hmac=aFzER8Y4wcWTrXVx2wVK
Sj10IqnygaF33gESj0WGDwI",
      time: "02:00 PM",
    },
  ],
  "2024-03-12": [
    {
      id: "event-3",
      title: "Yoga Session",
      description:
        "Join for a relaxing yoga session to reduce stress and improve
mindfulness. Suitable for all levels, focusing on gentle stretches.",
      imageUrl:

"https://fastly.picsum.photos/id/392/1920/1080.jpg?hmac=Fvbf7C1Rcozg8EccwYP
qsGkk_o6Bld2GQRDPZKWpd7g",
      time: "12:00 PM",
    },
    {
      id: "event-4",
      title: "Product Demo",
      description:
        "Demo of UI improvements and performance optimizations to gather
stakeholder feedback.",
      imageUrl:

"https://fastly.picsum.photos/id/249/1920/1080.jpg?hmac=cPMNdgGXRh6T_KhRMua
QjRtAx5cWRraELjtL2MHTfYs",
      time: "03:30 PM",
    },
  ],
  "2024-03-13": [
    {
      id: "event-5",
      title: "Client Meeting",
      description:
        "Review project progress, timeline adjustments, and outline roadmap
for next quarter with the client.",
      imageUrl:
```

```
"https://fastly.picsum.photos/id/908/1920/1080.jpg?hmac=MeG_oA1s75hHAL_4JzCioh6--zyFTWSCTxOhe8ugvXo",
      time: "11:30 AM",
    },
  ],
};

export default events;
```

In case you like the colours of the mock we provided, here are the AI-generated colours we used there:

Background Gradient:

- linear-gradient(135deg, #f6f8ff 0%, #eef1f9 100%)
    - Subtle light blue/gray gradient

Header Gradient:

- linear-gradient(to right, #3b82f6, #8b5cf6)
    - Blue to purple horizontal gradient

Active Buttons and Badges:

- linear-gradient(135deg, #4f46e5, #7c3aed)
    - Indigo to violet diagonal gradient

Or you can pick your own!