**Day 1 – Setup & Core “View” Flows**

1. **Repo & Tooling (1–2 hrs)**
   * Scaffold Vite + React + TypeScript.
   * Install: zustand, @tanstack/react-query, axios, react-router-dom, plus your chosen UI lib.
2. **Mock API (1 hr)**
   * Quick JSON-Server with endpoints:
     + POST /auth/login
     + GET /events
     + GET /events/:id/attendees
3. **Auth + Routing (3–4 hrs)**
   * Simple login form → React Query mutation → store token in Zustand.
   * Public /login vs protected /events routes.
4. **Events List Page (rest of day)**
   * Fetch + display basic table of events (name, date).
   * “View Details” link to /events/:id.

**Day 2 – “Create” & “Manage” Flows**

1. **Event Creation (2–3 hrs)**
   * Modal form: name, date, description, tickets.
   * React Query useMutation → POST → invalidate events list.
2. **Event Detail + Edit/Delete (3–4 hrs)**
   * Show event fields + “Edit” opens pre-filled creation modal; “Delete” with confirm.
   * Hook up PUT/DELETE via React Query.
3. **Attendees List (2–3 hrs)**
   * In detail view, fetch GET /events/:id/attendees; show simple list.
   * (Optional stretch: “Add Attendee” form.)

**Day 3 – Polish, Testing & Deploy**

1. **Styling & Responsiveness (3–4 hrs)**
   * Tidy up UI: ensure modals, tables, nav all look cohesive and work on mobile.
2. **Basic Tests (2 hrs)**
   * One or two Jest + React Testing Library tests:
     + Login flow
     + Event creation → appears in list
3. **Deployment (2 hrs)**
   * Push to GitHub + hook up Vercel/Netlify.
   * Set JSON-Server behind a simple proxy or deploy it (e.g. Heroku) or use a static mock data service.
4. **Final Smoke Check (1 hr)**
   * Walk through: login → create → edit → view attendees → delete.