Product Requirements & Specification Document

Project Name

Sportify - Sports Event Scheduler UI

Description

Sportify is a responsive frontend application for scheduling and managing sports events. The UI features event creation forms, interactive calendars, and event management tools. Built with React, advanced CSS, and Tailwind, Sportify targets startups and educational institutions seeking efficient event organization.

1. Goals & Objectives

Goal	Description
Event Scheduling	Allow users to create, edit, and delete sports events
Calendar Visualization	Display events in a calendar view
Responsive Design	Ensure usability across devices (desktop, tablet, mobile)
User-Friendly Interface	Provide intuitive navigation and clear feedback

2. Core Features

Feature	Description
Event Creation Form	Form to input event details (name, date, time, location, description, type)
Event List & Details	List of upcoming events with detail view
Calendar View	Monthly/weekly calendar with event markers
Edit/Delete Events	Ability to modify or remove existing events
Responsive Layout	Adaptive UI for various screen sizes using Tailwind
Validation & Feedback	Real-time form validation and user feedback

3. User Stories

As a…	I want to…	So that…
User	Create a new sports event	I can organize upcoming activities
User	View all scheduled events in a calendar	I can see event distribution over time
User	Edit or delete an event	I can manage changes or cancellations

User	Access the app on any device	I have flexibility and convenience

4. Functional Requirements

ID	Requirement
FR1	Users can create events via a form with required fields
FR2	Events are displayed in a list and calendar view
FR3	Users can edit or delete existing events
FR4	The UI is fully responsive
FR5	Form inputs are validated with clear error messages
FR6	Calendar view highlights days with events
FR7	Event details are viewable in a modal or detail pane

5. Non-Functional Requirements

ID	Requirement
NFR1	Built with React, JavaScript, HTML, CSS
NFR2	Uses Tailwind CSS for styling
NFR3	Loads in under 2 seconds on modern browsers
NFR4	Accessible (WCAG 2.1 AA compliance)
NFR5	Codebase is modular and maintainable

6. UI/UX Specifications

- **Navigation:** Top or side navigation bar for main sections (Calendar, Events, Create Event)
- **Forms:** Clean, minimal, with inline validation and clear labels
- **Calendar:** Interactive, supports month/week views, clickable event markers
- **Event List:** Card or table layout, sortable by date/type
- **Responsiveness:** Uses Tailwind breakpoints for adaptive layouts
- **Feedback:** Toasts or modals for success/error messages

7. Technical Specifications

Area	Specification
Framework	React (functional components, hooks)
Styling	Tailwind CSS, custom CSS for advanced components
State Mgmt	React Context or useState/useReducer

Routing	React Router (if multi-page)
Data Storage	In-memory (local state); mock API for demo
Validation	Custom or library-based (e.g., Yup)
Testing	Jest, React Testing Library (unit/component tests)

8. Acceptance Criteria

- □ Users can create, edit, and delete events
- □ Events are visible in both list and calendar views
- □ UI is responsive and accessible
- □ Form validation prevents invalid submissions
- □ All core features function on latest Chrome, Firefox, Safari

9. Out of Scope

- User authentication
- Backend integration (beyond mock/demo)
- Notifications/reminders
- Payment or ticketing features

10. Sample UI Structure (Pseudocode)

```
<App>
  <NavBar />
  <Main>
    <CalendarView />
    <EventList />
    <EventForm />
    <EventDetailModal />
  </Main>
</App>
```

11. Milestones & Timeline

Milestone	Target Date
UI Wireframes	Week 1
Core Components	Week 2
Calendar Integration	Week 3
Responsive Design	Week 4

Testing & QA	Week 5
Final Review & Delivery	Week 6

12. Stakeholders

Role	Name/Group
Product Lead	Startup/Education
Developers	Frontend Team
Designers	UI/UX Team

End of Document