📄 Sydney Events Website – Project Documentation

Tech Stack: React, Vite, Tailwind CSS, Ticketmaster API
Project Goal: Build a live, responsive event-listing website for Sydney, Australia, with automatic updates and email opt-in functionality.

🧩 Project Overview

This project is a modern, responsive web application that dynamically lists events happening in **Sydney, Australia**. It fetches real-time data from the **Ticketmaster API**, refreshes regularly, and allows users to **opt-in with their email** before being redirected to the ticket booking page. The application is fully responsive and built with industry-standard tools and best practices.

⚙️ Tech Stack

Area	Technology
Frontend Framework	React + Vite
Styling	Tailwind CSS
API Integration	Ticketmaster API
State Management	React Hooks
Icons	React Icons
Deployment	Vercel (view live)

✅ Features Implemented

1. 🔄 Real-Time Event Data

- Integrated with **Ticketmaster API** to fetch events for Sydney, Australia.
- Data refreshes every **5 minutes** automatically.
- Implemented error handling and fallback UI for network issues.

2. 📱 Responsive User Interface

- Fully responsive grid layout (mobile: 1 column, tablet: 2, desktop: 3).
- Visually appealing **gradient header** and animated card effects.
- Truncation of text for uniformity and design consistency.
- Includes **skeleton loader** for better user experience during data fetch.

3. 🔍 Search and Filter Functionality

- **Real-time search** for event names and venues.
- **Category filters**: All, Music, Sports, Arts, Family.
- Combined search and filter logic to improve discoverability.

4. 📦 Event Card Design

Each event card includes:

- Event image with a fallback
- Truncated title and venue for consistency
- Date and location with corresponding icons
- Interactive hover effects and responsive styling

5. 📧 Email Collection + Redirection

- Modal-based form that collects user email before redirection

- Email validation + newsletter opt-in checkbox
- Secure redirection to the event’s official page
- Email preference stored locally for enhanced UX

📁 Codebase Structure

```
frontend/
├── src/
│   ├── components/
│   │   ├── EmailModal.jsx      // Handles email input and redirection
│   │   └── EventCard.jsx       // UI for each event
│   ├── pages/
│   │   └── EventsPage.jsx      // Main page layout
│   ├── services/
│   │   └── eventService.js     // API communication with Ticketmaster
│   ├── App.jsx                 // Root component
│   └── index.css               // Global Tailwind CSS styles
│
```

🚀 Setup Instructions

Prerequisites:

- Node.js & npm
- A valid Ticketmaster API key

Installation:

1. **Clone the project:**

```
        git clone https://github.com/your-username/sydney-events
cd sydney-events
```

2. **Install dependencies:**

```
npm install
```

3. **Set up environment variables:**
   Create a `.env` file and add:

   ```
   VITE_TICKETMASTER_API_KEY=your_api_key_here
   ```

4. **Start development server:**

   ```
   npm run dev
   ```

🔐 Environment Variables

```
VITE_TICKETMASTER_API_KEY=your_api_key_here
```

📌 Known Limitations

- Currently using **localStorage** for email collection (no backend).
- Event data limited to **Ticketmaster API** only.
- No persistent user preferences or bookmarks.
- Limited filter/sort options at this stage.

🔮 Suggested Improvements (Future Scope)

🔗 Backend Integration

- Add backend (e.g., Node.js, Express, or FastAPI) for secure email storage.
- Integrate database (MongoDB/Firebase) to persist user data.

📊 Feature Enhancements

- Add sorting options (e.g., by popularity/date).
- Include ticket price info, full event descriptions, and organizer data.
- Implement favorites/bookmarked events for logged-in users.

📱 Mobile Experience

- Add mobile-first features like swipe filters, collapsible cards, and fast loading.

🧠 Advanced Features

- Google Maps for venue locations.
- Social media sharing & calendar sync.
- Event recommendation system (AI-driven).

🔒 Accessibility & Quality

- Improve keyboard navigation and screen reader support.
- Add unit and integration tests using **Jest** and **React Testing Library**.
- Add analytics and error tracking tools (e.g., Google Analytics, Sentry).

🛣️ Roadmap

Phase	Goal
Phase 1	Backend for email collection
Phase 2	Persistent user preferences and admin dashboard
Phase 3	Event bookmarking, price filters, and social features
Phase 4	Mobile app (React Native or Flutter)
Phase 5	Performance optimization and analytics

✅ Summary

This project demonstrates the ability to:

- Build a modern, scalable, and responsive frontend using open-source tools.
- Integrate with a real-world API (Ticketmaster) and handle dynamic data.
- Apply frontend best practices like state management, component reuse, and responsive design.
- Implement practical user flows including search, filters, modals, and redirection with validations.

Thank you for reviewing this assignment. I look forward to any feedback or questions.
– Abhishek Madoliya