# Phase 6: User Interface Development — Event Management System

## Objective

Build a modern, responsive Salesforce UI (Lightning) for event creators, coordinators, attendees, and marketing — using App Builder, Record Pages, LWC, and Apex integrations to deliver a polished UX and fast workflows.

## 1. Lightning App Builder

Purpose: Assemble apps and pages (App Page, Home Page, Record Page) with declarative components.

Guidance:

- Create a dedicated Event Management App with custom navigation (Events, Attendees, Venues, Reports).  
- Use App Page for event dashboards (metrics, upcoming events, registration trends).  
- Restrict app visibility via Profiles/Permission Sets.

Deliverables:

- Named App: Event Management  
- App-level branding (logo, color), default home tab.  
- App navigation items: Event\_\_c, Attendee\_\_c, Venue\_\_c, Event Reports, Setup.

## 2. Record Pages

Purpose: Custom layouts for Event and Attendee records.

Design:

- Use record page variants by profile (Organizer vs. Attendee Support).  
- Sections: Header, Event Details, Schedule, Attendees, Tasks, Files, Related Actions.

## 3. Tabs

Purpose: Provide clear top-level navigation inside the App.

Examples:  
- Standard tabs: Home, Events, Attendees, Venues, Reports, Dashboards.  
- Custom Lightning Page tab for the Event Dashboard.

## 4. Home Page Layouts

Purpose: Provide role-specific summaries and shortcuts.

Components:  
- Event Summary Chart  
- Quick Actions  
- Recent Items  
- KPI Tiles

Deliverables:  
- Two home pages: Organizer Home, Support Home.

## 5. Utility Bar

Purpose: Persistent tools accessible from any app page.

Utilities:  
- Quick Create Event  
- Event Search  
- Notification Center  
- Countdown Timer

## 6. Lightning Web Components (LWC)

Purpose: Build reusable, fast UI components.

Example: eventCard component.

Deliverables:  
- LWC library: eventCard, eventList, attendeeForm, eventDashboard, fileUploader.

## 7. Apex with LWC

Purpose: Use Apex back-end methods from LWC for complex logic or SOQL not available via wire.

Example: EventController.cls with @AuraEnabled methods for event retrieval and registration.

## 8. Events in LWC

Purpose: Enable communication between components using custom events or LMS.

Types:  
- DOM/Custom events  
- Lightning Message Service (LMS)  
- Platform Events

## 9. Wire Adapters

Purpose: Declarative data retrieval in LWC using cacheable Apex methods.

Examples:  
- getRecord for record data  
- getUpcomingEvents for Apex wired data.

## 10. Imperative Apex Calls

Purpose: Call non-cacheable Apex (writes, DML, callouts) from LWC.

Example: registerAttendee Apex method called imperatively to register new attendees.

## 11. Navigation Service

Purpose: Navigate programmatically in Lightning Experience and mobile.

Examples:  
- Navigate to record view page  
- Navigate to named page or app.