**Project Title:** Smart Event Management System with Automated Volunteer Assignment

**Phase 1: Problem Understanding & Industry Analysis**

👉 **Goal:** Understand why this system is needed.

**Requirement Gathering**

* Talk to stakeholders (Event Managers, Clients, Vendors, Volunteers).
* Example requirements:
  + Track event details (venue, date, budget, attendees).
  + Manage client registrations and bookings.
  + Allocate vendors (catering, logistics, stage setup).
  + Automate approvals for high-budget events.
  + Generate event performance & revenue reports.

**Stakeholder Analysis**

* **Admin** → manages overall Salesforce setup.
* **Event Manager** → creates & manages events.
* **Vendors** → provide services (catering, lighting, decoration).
* **Clients** → request events, register as attendees.
* **Finance Team** → monitors payments & budgets.

**Business Process Mapping**

Flow:  
Client requests event → Event Manager creates event → Vendor allocation → Approval (if budget exceeds threshold) → Notifications sent → Event executed → Reports generated.

**Industry-specific Use Case Analysis**

* In event management, **deadlines and budgets** are critical.
* Vendors have dependencies (venue booked before catering).
* Automation reduces manual scheduling conflicts.

**AppExchange Exploration**

* Checked for “Event Management” apps. While some exist, this project is a **custom Salesforce solution** for learning & control.

**Phase 2: Org Setup & Configuration**

👉 **Goal:** Prepare Salesforce developer environment.

* **Salesforce Edition:** Developer Org (free).
* **Company Profile Setup:** Set timezone (IST), currency (INR/USD).
* **Business Hours:** 9am–8pm (event operations).
* **Users Setup:** Event Manager, Vendor, Finance Team.
* **Profiles:**
  + Event Manager → Full access to Events & Reports.
  + Vendors → Restricted access (only allocated events).
* **Roles:** Hierarchy → Manager > Event Manager > Vendors.
* **Permission Sets:** For special access (e.g., Reports, Dashboards).
* **OWD (Org-Wide Defaults):**
  + Event object: Public Read/Write.
  + Vendor object: Private.
* **Sharing Rules:** Allow Event Managers to see each other’s events.

**Phase 3: Data Modeling & Relationships**

👉 **Goal:** Build database structure.

**Objects & Fields**

* **Standard Objects:** Contact (Clients/Attendees), Account (Corporate Clients).
* **Custom Objects:**
  + Event (Event Name, Date, Venue, Budget, Status).
  + Vendor (Service Type, Contact Info, Availability).
  + Booking (Client, Event, Payment).

**Relationships**

* **Event ↔ Vendor:** Many-to-Many (via junction object → EventVendor).
* **Event ↔ Client:** Lookup.
* **Event ↔ Booking:** Master-Detail.

**Record Types**

* Event Types → Corporate Event, Wedding, Concert, Conference.

**Page Layouts & Compact Layouts**

* Event page → Vendors & Bookings.
* Booking page → Linked Event & Client.

**Phase 4: Process Automation (Admin)**

👉 **Goal:** Automate workflows.

* **Validation Rules:**
  + Event Date ≥ Today.
  + Budget > 0.
* **Approval Process:**
  + If Budget > ₹5,00,000 → requires Finance approval.
* **Flow Builder:**
  + Record-triggered Flow → Auto-calculate total cost.
  + Screen Flow → New Booking form.
* **Email Alerts:**
  + Notify Client after booking approval.
  + Notify Vendor when assigned to event.
* **Tasks:**
  + Auto-create task for Event Manager to review vendor setup.

**Phase 5: Apex Programming (Developer)**

👉 **Goal:** Add advanced business logic.

* **Apex Trigger:** Prevent double-booking of the same vendor on overlapping event dates.
* **Service Class:** EventService → calculate event revenue, allocate vendors.
* **SOQL Queries:** Fetch available vendors WHERE status = Available AND date not overlapping.
* **Batch Apex:** Night job → Send reminders for next-day events.
* **Future Methods / Callouts:** Integration with external payment gateway (simulation).
* **Test Classes:** Validate triggers & automation.

**Phase 6: User Interface Development**

👉 **Goal:** Build a user-friendly experience.

* **Lightning App Builder:** Create “Smart Event Management” app.
* **Tabs:** Events, Vendors, Clients, Bookings.
* **Home Page:** Dashboard (Upcoming Events, Top Clients, Pending Approvals).
* **LWC Components:**
  + Search Events by Date/Location.
  + Vendor Availability Checker.
* **Navigation Service:** Redirect to Event details after booking.

**Phase 7: Integration & External Access**

👉 **Goal:** Connect external systems.

* **Named Credentials:** For secure API calls (payment gateway).
* **REST Callouts:** Check external vendor availability API (simulated).
* **Platform Events:** Notify Vendors in real time when assigned.
* **Change Data Capture:** Track updates in Event status.

**Phase 8: Data Management & Deployment**

👉 **Goal:** Manage data and move changes.

* **Data Import Wizard:** Import 50 demo Clients & Vendors.
* **Data Loader:** Bulk upload Bookings.
* **Duplicate Rules:** Prevent duplicate Client entries.
* **Data Export:** Weekly backup.
* **Deployment:** VS Code + SFDX CLI for changes.

**Phase 9: Reporting & Dashboards**

👉 **Goal:** Provide insights to stakeholders.

* **Reports:**
  + Events by Month.
  + Revenue by Event Type.
  + Vendor Utilization.
* **Dashboards:**
  + Event Manager Dashboard (Upcoming Events).
  + Finance Dashboard (Revenue vs Budget).
* **Dynamic Dashboards:** Role-based visibility.

**Phase 10: Final Presentation & Demo**

👉 **Goal:** Deliver project professionally.

* **Pitch Presentation:** Problem → Solution → Benefits.
* **Demo Walkthrough:** Event creation → Vendor assignment → Approval → Reports.
* **Handoff Documentation:**
  + Project design doc.
  + User guide.
* **Portfolio Showcase:** Add to GitHub & LinkedIn.