

# PHASE 9: REPORTS, DASHBOARDS & SECURITY

**Project:** EventEase – AI-Powered Event Registration System

**Phase:** 9 – Reports, Dashboards & Security

**Developer:** Yash Lawankar

**Date:** September 2025

**Mentor Review Document**

---

## EXECUTIVE SUMMARY

Phase 9 delivers production-grade analytics and role-based security for EventEase. Key achievements include:

- **Reports:** 3 core reports created and validated for registrations, trends, and revenue.
  - **Live Monitoring:** 2 Lightning Web Components (LWCs) deployed on Home for real-time insights.
  - **Security:** Role-based permission sets, Apex class access, and field-level security (FLS) implemented.
  - **Outcome:** Stakeholders gain real-time visibility with proper access control, reducing navigation and manual checks.
- 

## OBJECTIVES

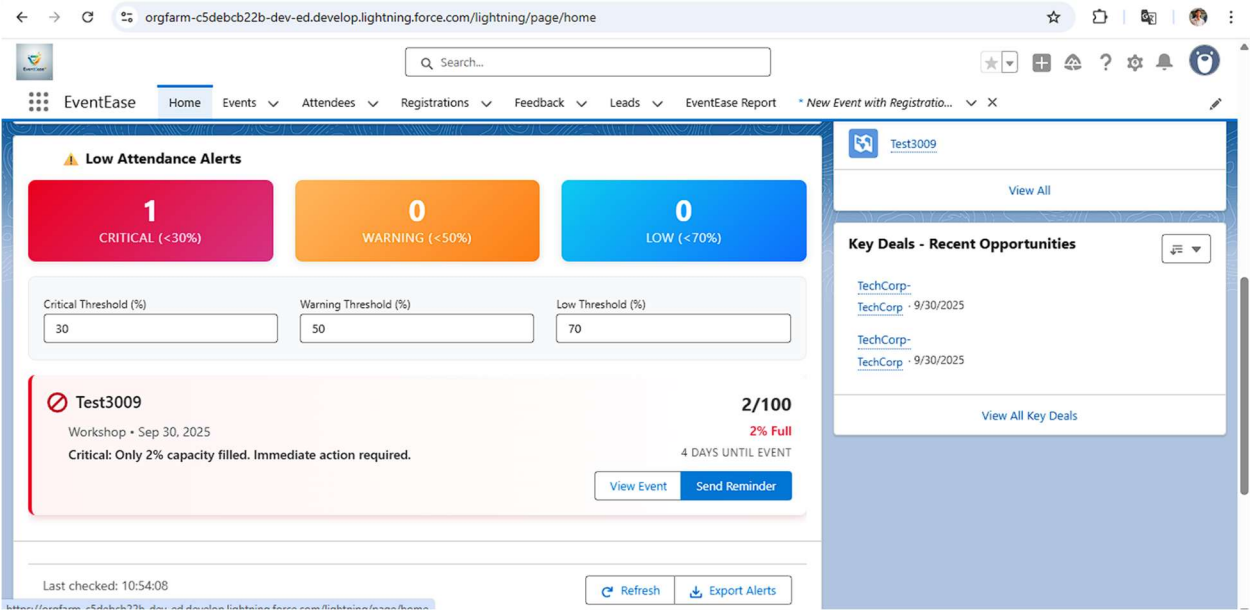
1. Provide actionable insights via curated reports.
  2. Enable at-a-glance monitoring on the Home page through LWCs.
  3. Enforce role-based access control for Admins, Managers, and Analysts.
-

# SCOPE & ASSUMPTIONS

- **Scope:** Reports, Home page monitoring LWCs, and Permission Sets.
- **Assumptions:** Core objects (Event\_\_c, Attendee\_\_c, Registration\_\_c, Feedback\_\_c) and Email/Trigger logic are production-ready.
- **Prerequisite:** Report Types configured and deployed before report deployment.

# DELIVERABLES

## 1. Reports (Production-Ready)



Report Name	Type	Key Details
Registrations per Event	Summary	Total & confirmed registrations, event details (Type, Date, Capacity, Status), status/payment breakdown

Report Name	Type	Key Details
Attendance Trends Matrix	Matrix	Registration count by week vs. event; last 90-day trend view
Popular Events & Revenue	Summary	Revenue per event, attendance totals, average rating, response count; sorted by revenue descending

Supporting Report Types:

- Event with Registrations
- Event with Registrations and Feedback

orgfarm-c5debc22b-dev-ed.develop.lightning.force.com/lightning/r/Report/00OgL000005PEGDUA4/view?queryScope=userFolders

EventEase

Home

Events

Attendees

Registrations

Feedback

Leads

EventEase Report

New Event with Registratio...

Report: Event with Registrations

New Event with Registrations Report

Event Ease Report

Enable Field Editing

Add Chart

Edit

Total Records

14

	Event Name	Event Type	Event Date	Capacity	Event Status	Status	Registration Date	Payment Status	Amount
1	Online AI Workshop 2024	Workshop	9/23/2025, 12:00 PM	19	Registration Closed	Confirmed	9/20/2025, 5:00 PM	Pending	-
2	Online AI Workshop 2024	Workshop	9/23/2025, 12:00 PM	19	Registration Closed	Confirmed	9/20/2025, 5:00 PM	Pending	-
3	Online AI Workshop 2024	Workshop	9/23/2025, 12:00 PM	19	Registration Closed	Confirmed	9/20/2025, 5:00 PM	Pending	-
4	QA Test CConference 22 sep 25	-	9/22/2025, 9:00 PM	19	Active	Confirmed	9/21/2025, 6:50 AM	Paid	\$3,000.00
5	QA Test CConference 22 sep 25	-	9/22/2025, 9:00 PM	19	Active	Confirmed	9/21/2025, 7:32 AM	Paid	\$3,000.00
6	Online AI Workshop 2024	Workshop	9/23/2025, 12:00 PM	19	Registration Closed	Confirmed	9/21/2025, 7:35 AM	Paid	\$3,000.00
7	Online AI Workshop 2024	Workshop	9/23/2025, 12:00 PM	19	Registration Closed	Confirmed	9/21/2025, 7:44 AM	Paid	\$3,000.00
8	Online AI Workshop 2024	Workshop	9/23/2025, 12:00 PM	19	Registration Closed	Confirmed	9/21/2025, 8:22 AM	Paid	\$3,000.00
9	QA Test CConference 22 sep 25	-	9/22/2025, 9:00 PM	19	Active	Confirmed	9/21/2025, 5:00 PM	Paid	\$3,000.00
10	Online AI Workshop 2024	Workshop	9/23/2025, 12:00 PM	19	Registration Closed	Confirmed	9/21/2025, 8:45 PM	Paid	\$3,000.00
11	TestIP	Conference	9/23/2025, 12:00 PM	100	Registration Open	Cancelled	9/22/2025, 11:11 AM	Paid	\$1,500.00
12	TestIP	Conference	9/23/2025, 12:00 PM	100	Registration Open	Confirmed	9/22/2025, 11:34 AM	Paid	\$1,500.00

2. Live Monitoring (Home Widgets)

Widget	Features
Registrations Counter (LWC)	Live totals (Total / Confirmed / Pending), event filter, auto-refresh (30s), capacity progress with color thresholds

Widget	Features
Low Attendance Alerts (LWC)	Highlights events below target thresholds (<30% critical); auto-refresh periodically

### 3. Security (Role-Based)

Permission Sets Deployed & Assigned:

<input type="checkbox"/>	<a href="#">Del</a>   <a href="#">Clone</a>	<a href="#">EventEase Admin Permissions</a>	<a href="#">Save...</a>	Administrative permissions for EventEase system manage...
<input type="checkbox"/>	<a href="#">Del</a>   <a href="#">Clone</a>	<a href="#">EventEase Analyst</a>		Analyst permissions for EventEase: read-only access and ...
<input type="checkbox"/>	<a href="#">Del</a>   <a href="#">Clone</a>	<a href="#">EventEase Full Access</a>		
<input type="checkbox"/>	<a href="#">Del</a>   <a href="#">Clone</a>	<a href="#">EventEase Manager</a>		Manager permissions for EventEase: manage registrations...

- EventEase\_Admin\_Permissions
- EventEase\_Manager
- EventEase\_Analyst
- EventEase\_Full\_Access (legacy full access)

## Apex Class Access:

- LowAttendanceController

The screenshot shows the Salesforce Setup interface for the 'LowAttendanceController' Apex Class. The browser address bar displays the URL: `orgfarm-c5debcb22b-dev-ed.develop.lightning.force.com/lightning/setup/ApexClasses/page?address=%2F01pgL000005rNA5`. The left sidebar shows the 'Setup' menu with 'Apex Classes' selected under 'Custom Code'. The main content area displays the 'Apex Class Detail' for 'LowAttendanceController'. The class is owned by 'Yash Lawankar' and was created on 9/25/2025 at 7:01 PM. The status is 'Active'. The 'Class Body' tab is selected, showing the following code:

```

1 public with sharing class LowAttendanceController {
2
3     @AuraEnabled(cacheable=true)
4     public static List<AttendanceAlert> getLowAttendanceAlerts(Integer criticalThreshold, Integer warningThreshold, Integer lowThreshold) {
5         try {
6             List<Event__c> events = [
7                 SELECT Id, Name, Event_Type__c, Event_Date__c, Capacity__c,
8                   (SELECT Id, Status__c FROM Registrations__r
9                    WHERE Status__c IN ('Confirmed', 'Approved'))
10             FROM Event__c
11             WHERE Event_Date__c >= TODAY
12             AND Event_Date__c <= NEXT_N_DAYS:90
13             AND Status__c != 'Cancelled'
14             AND Capacity__c > 0
15             ORDER BY Event_Date__c ASC
16         ];
17         List<AttendanceAlert> alerts = new List<AttendanceAlert>();
18         for (Event__c event : events) {
19             List<Registration__c> registrations = [SELECT Id FROM Registrations__r WHERE Event__c = :event.Id];
20             Integer criticalCount = 0;
21             Integer warningCount = 0;
22             Integer lowCount = 0;
23             for (Registration__c reg : registrations) {
24                 if (reg.Status__c == 'Confirmed') {
25                     criticalCount++;
26                 } else if (reg.Status__c == 'Approved') {
27                     warningCount++;
28                 } else if (reg.Status__c == 'On Hold') {
29                     lowCount++;
30                 }
31             }
32             alerts.add(new AttendanceAlert(event.Id, event.Name, criticalCount, warningCount, lowCount));
33         }
34         return alerts;
35     }
36 }

```

- RegistrationsCounterController

The screenshot shows the Salesforce Setup interface for the 'RegistrationsCounterController' Apex Class. The browser address bar displays the URL: `orgfarm-c5debcb22b-dev-ed.develop.lightning.force.com/lightning/setup/ApexClasses/page?address=%2F01pgL000005rQ1V`. The left sidebar shows the 'Setup' menu with 'Apex Classes' selected under 'Custom Code'. The main content area displays the 'Apex Class Detail' for 'RegistrationsCounterController'. The class is owned by 'Yash Lawankar' and was created on 9/25/2025 at 8:19 PM. The status is 'Active'. The 'Class Body' tab is selected, showing the following code:

```

1 public with sharing class RegistrationsCounterController {
2
3     @AuraEnabled(cacheable=true)
4     public static List<RegistrationStats> getRegistrationStats(String eventId) {
5         try {
6             String query = 'SELECT Id, Status__c FROM Registration__c';
7             List<String> conditions = new List<String>();
8
9             if (String.isNotBlank(eventId)) {
10                 conditions.add('Event__c = :eventId');
11             }
12
13             if (!conditions.isEmpty()) {
14                 query += ' WHERE ' + String.join(conditions, ' AND ');
15             }
16             List<Registration__c> registrations = [SELECT Id, Status__c FROM Registration__c];
17             List<RegistrationStats> stats = new List<RegistrationStats>();
18             for (Registration__c reg : registrations) {
19                 Integer confirmedCount = 0;
20                 Integer approvedCount = 0;
21                 Integer onHoldCount = 0;
22                 if (reg.Status__c == 'Confirmed') {
23                     confirmedCount++;
24                 } else if (reg.Status__c == 'Approved') {
25                     approvedCount++;
26                 } else if (reg.Status__c == 'On Hold') {
27                     onHoldCount++;
28                 }
29                 stats.add(new RegistrationStats(reg.Id, reg.Event__c, confirmedCount, approvedCount, onHoldCount));
30             }
31             return stats;
32         } catch (Exception e) {
33             return new List<RegistrationStats>();
34         }
35     }
36 }

```

## Field-Level Security (FLS) Hardened:

- Registration\_\_c.Payment\_Status\_\_c
  - Event\_\_c.Meeting\_Link\_\_c
  - Event\_\_c.Venue\_\_c
- 

## TECHNICAL IMPLEMENTATION

### Report Types & Reports

- Child fields exposed in report type layout to fix "invalid report type".
- Final reports retrieved from org to sync XML tokens prior to deployment.

### LWCs & Apex

#### RegistrationsCounterController.cls

- Methods: getRegistrationStats(eventId), getEventCounters(eventId), getEventOptions()
- Logic: Counts confirmed/approved registrations, calculates capacity usage, returns event options for filtering.

#### registrationsCounter (LWC)

- Auto-refresh (30s), event filter, color-coded capacity bar (green/yellow/orange/red).

#### lowAttendanceAlerts (LWC)

- Categorizes alerts (Critical/Warning/Low) based on attendance thresholds.

### Security (Permission Sets)

- Correct metadata schema used (classAccesses) for Apex access.
- Field permissions grouped properly; FLS applied for sensitive fields.

- User permissions: RunReports, ViewPublicReports/Dashboards, SubscribeToLightningReports/Dashboards where applicable.
- 

## DEPLOYMENT SUMMARY (KEY COMMANDS)

### Deploy/Sync Report Types & Reports:

```
sf project deploy start --source-dir force-app/main/default/reportTypes
sf project deploy start --source-dir force-app/main/default/reports/EventEase_Reports
sf project retrieve start --metadata "Report:EventEase_Reports/Registrations_per_Event"
sf project retrieve start --metadata "Report:EventEase_Reports/Attendance_Trends_Matrix"
sf project retrieve start --metadata "Report:EventEase_Reports/Popular_Events_Revenue_Report"
sf project retrieve start --metadata "ReportType:Event_with_Registrations"
```

### Deploy LWCs & Controllers:

```
sf project deploy start --source-dir force-app/main/default/classes/RegistrationsCounterController.cls --wait 10
sf project deploy start --source-dir force-app/main/default/lwc/registrationsCounter --wait 10
```

### Deploy & Assign Permission Sets:

```
sf project deploy start --source-dir force-app/main/default/permissionsets --wait 10
sf org assign permset --name EventEase_Admin_Permissions
sf org assign permset --name EventEase_Manager
sf org assign permset --name EventEase_Analyst
```

---

## VALIDATION & TESTING

- **Reports:** Data loads correctly; groupings, filters, totals/subtotals verified.
- **LWCs:**

- Registrations Counter updates on filter change; auto-refresh works; threshold colors accurate.
- Low Attendance Alerts display appropriate categories and update periodically.
- **Security:** Non-admin users (Manager/Analyst) can:
  - View LWCs on Home.
  - Open/run reports in EventEase\_Reports folder.
  - Subscribe to reports/dashboards (if permitted).

## SECURITY IMPLEMENTATION DETAILS

### Permission Sets

Role	Access Details
Admin	Full CRUD on core objects; create/run reports/dashboards; Apex access; FLS for sensitive fields
Manager	Read Events/Attendees, Create/Edit Registrations/Feedback; analytics permissions; Apex access
Analyst	Read-only access to core objects; analytics permissions; Apex access

### FLS

- Sensitive fields (Payment Status, Meeting Link, Venue) secured by role.

### Folder Sharing (Recommended)

- EventEase\_Reports: Managers (Editor), Analysts (Viewer), Admin as Manager



## USER GUIDE (ACCESS)

- **Reports:** App → Reports → Folders → EventEase\_Reports → Open reports
- **Home Widgets:** Registrations Counter & Low Attendance Alerts visible after App Builder placement
- **Dashboard (Optional):** Create “EventEase Dashboard” from 3 reports → Add component on Home → Save & Activate
- **Subscriptions:** From report/dashboard → Subscribe → Select schedule & recipients

## ISSUES RESOLVED IN PHASE 9

Issue	Cause	Fix
Invalid Report Type	Fields not exposed in report type layout	Added required fields; retrieved correct XML; redeployed
LWC Apex Reference Error	LWC deployed before Apex class	Deployed RegistrationsCounterController prior to LWC
Permission Set Parsing	Wrong tag name, misplaced fieldPermissions	Corrected classAccesses; grouped FLS; deployed and assigned successfully

## ARTIFACTS & PATHS (REPO)

### Report Types:

- force-app/main/default/reportTypes/Event\_with\_Registrations.reportType-meta.xml
- force-app/main/default/reportTypes/Event\_with\_Registrations\_and\_Feedback.reportType-meta.xml

**Reports:**

- force-app/main/default/reports/EventEase\_Reports/Registrations\_per\_Event.report-meta.xml
- force-app/main/default/reports/EventEase\_Reports/Attendance\_Trends\_Matrix.report-meta.xml
- force-app/main/default/reports/EventEase\_Reports/Popular\_Events\_Revenue\_Report.report-meta.xml

**LWCs:**

- force-app/main/default/lwc/registrationsCounter/
- force-app/main/default/lwc/lowAttendanceAlerts/

**Apex Controllers:**

- force-app/main/default/classes/RegistrationsCounterController.cls
- force-app/main/default/classes/LowAttendanceController.cls

**Permission Sets:**

- force-app/main/default/permissionsets/EventEase\_Admin\_Permissions.permissionset-meta.xml
- force-app/main/default/permissionsets/EventEase\_Manager.permissionset-meta.xml
- force-app/main/default/permissionsets/EventEase\_Analyst.permissionset-meta.xml
- force-app/main/default/permissionsets/EventEase\_Full\_Access.permissionset-meta.xml