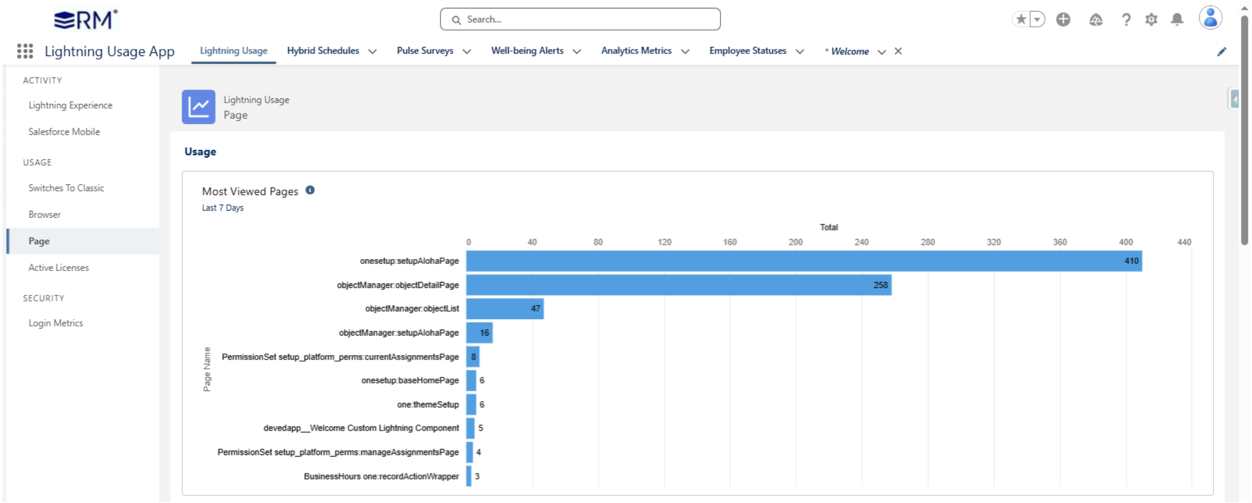
ERM Hybrid Workforce Project – Phase 6

Phase 6: User Interface Development

---

1. Lightning App Builder

- Created a dedicated **ERM Hybrid Workforce Lightning App**.
- Added custom branding (app name, logo, description).
- Enabled **standard navigation** for a user-friendly top bar.
- Configured **profiles** (HR Manager, Team Manager, Employee) to access the app.



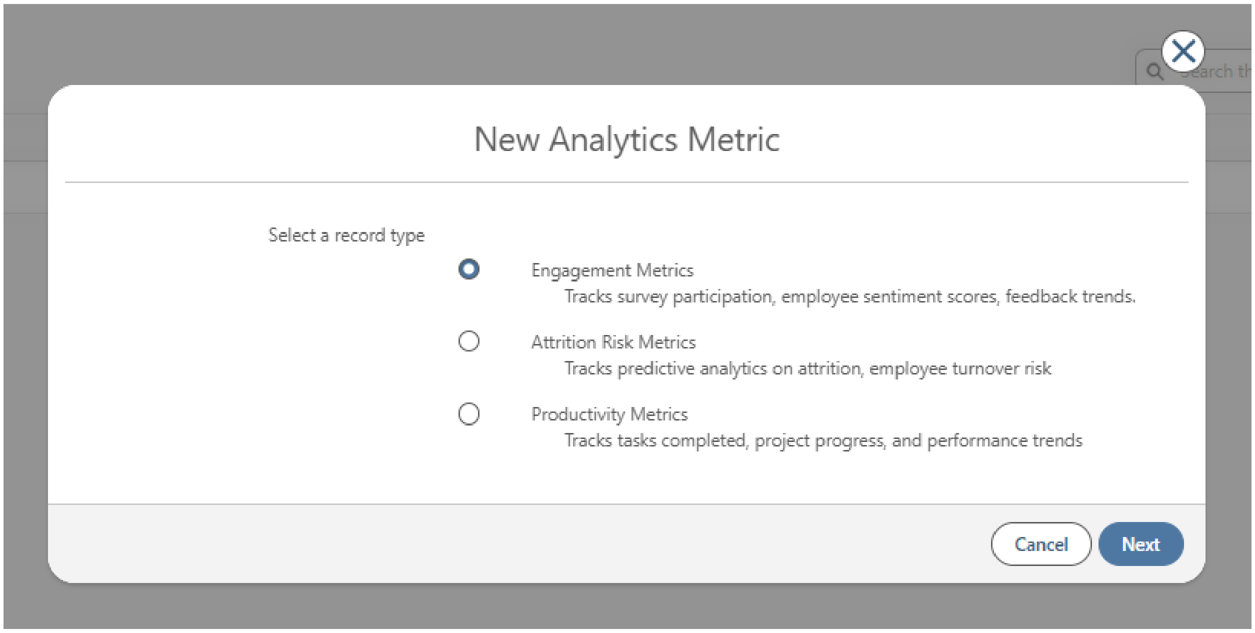
---

2. Record Pages

Customized **Record Pages** for each major object:

- **Employee__c Record Page** → Employee details, related schedules, surveys, and well-being alerts.
- **Hybrid_Schedule__c Record Page** → Displays schedule info, work mode, dates, and employee lookup.
- **Pulse_Survey__c Record Page** → Includes sentiment score, feedback, and related employee information.

- **Well_Being_Alert__c Record Page** → Shows alert severity, type, status, and trigger date.
- **Analytics_Metric__c Record Page** → Configured to display metric trends and insights



New Analytics Metric

Select a record type

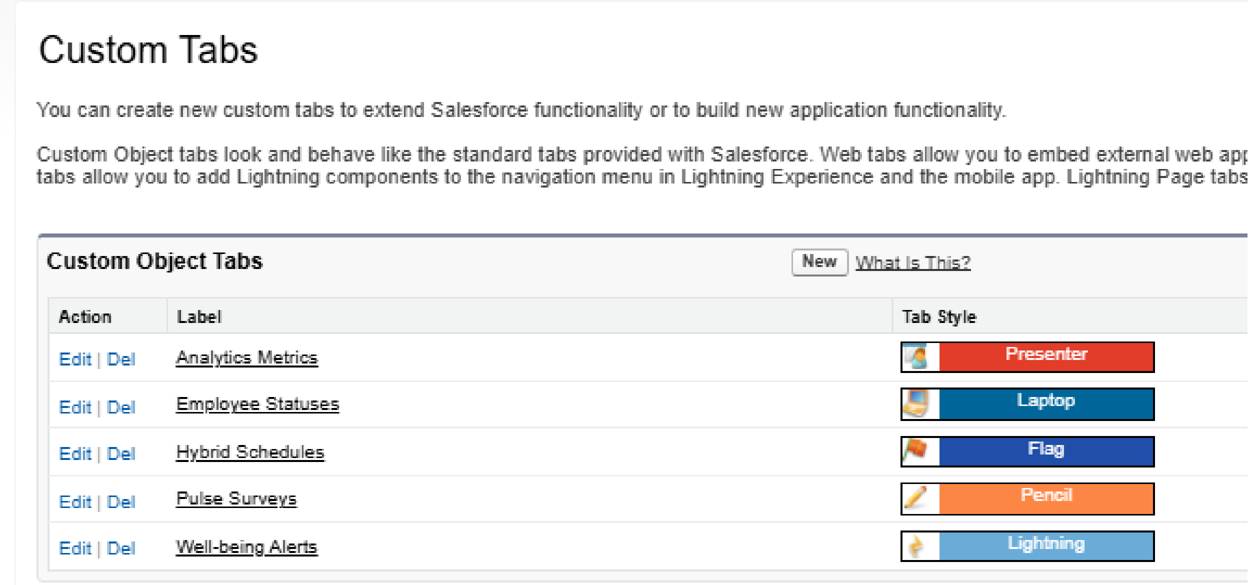
● Engagement Metrics
Tracks survey participation, employee sentiment scores, feedback trends.

○ Attrition Risk Metrics
Tracks predictive analytics on attrition, employee turnover risk

○ Productivity Metrics
Tracks tasks completed, project progress, and performance trends

3. Tabs

Added **navigation tabs** for all major project objects:



Custom Tabs

You can create new custom tabs to extend Salesforce functionality or to build new application functionality.

Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web app
tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs

Custom Object Tabs

Action	Label	Tab Style
Edit | Del	Analytics Metrics	Presenter
Edit | Del	Employee Statuses	Laptop
Edit | Del	Hybrid Schedules	Flag
Edit | Del	Pulse Surveys	Pencil
Edit | Del	Well-being Alerts	Lightning

- Employees

- Hybrid Schedules
- Pulse Surveys
- Well-being Alerts
- Analytics Metrics

---

4. Home Page Layouts

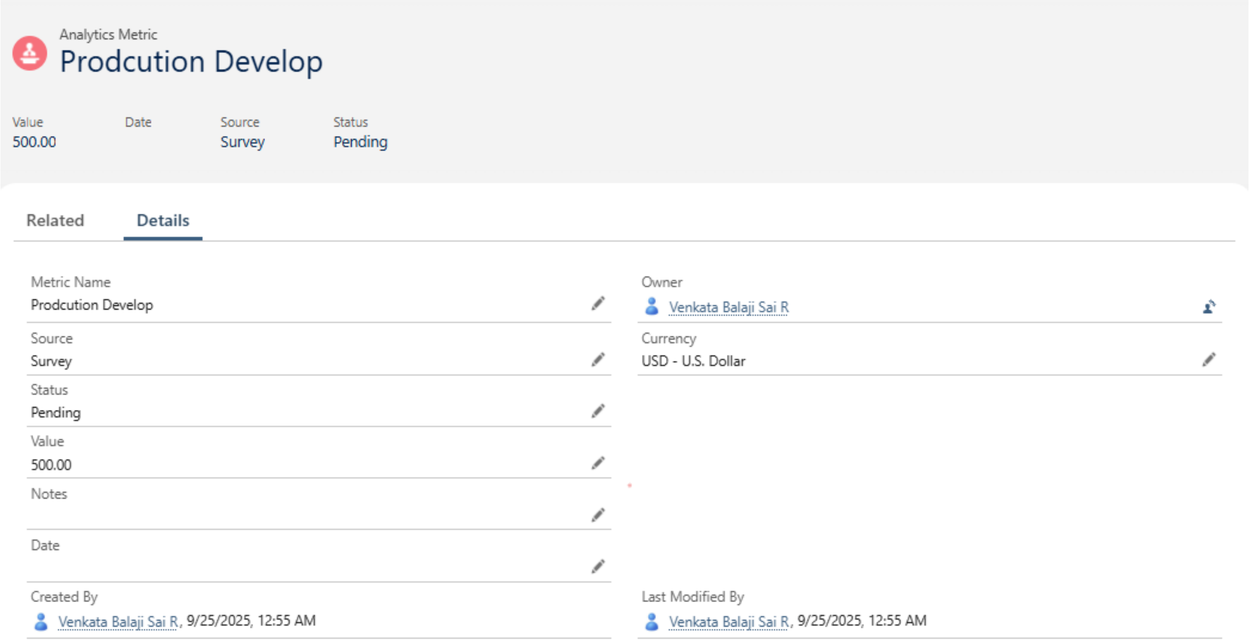
Designed role-based **Home Pages**:

- **Managers** → Pending schedule approvals, team well-being alerts, engagement dashboards.
- **HR** → Org-wide dashboards, pending surveys, and alerts monitoring.
- **Employees** → My schedule, survey submissions, and personal well-being panel.

---

5. Utility Bar

- Added quick actions:
  - Submit Pulse Survey
  - Create Hybrid Schedule
  - Critical Alerts Panel
  - Knowledge Base Search



Analytics Metric

Prodcution Develop

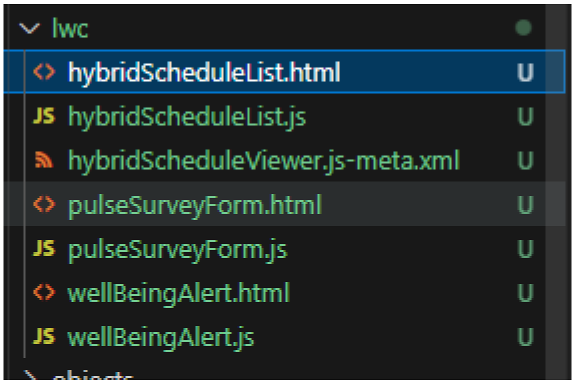
Value	Date	Source	Status
500.00		Survey	Pending

Metric Name Prodcution Develop	Owner Venkata Balaji Sai R
Source Survey	Currency USD - U.S. Dollar
Status Pending	
Value 500.00	
Notes	
Date	
Created By Venkata Balaji Sai R, 9/25/2025, 12:55 AM	Last Modified By Venkata Balaji Sai R, 9/25/2025, 12:55 AM

6. Lightning Web Components (LWC)

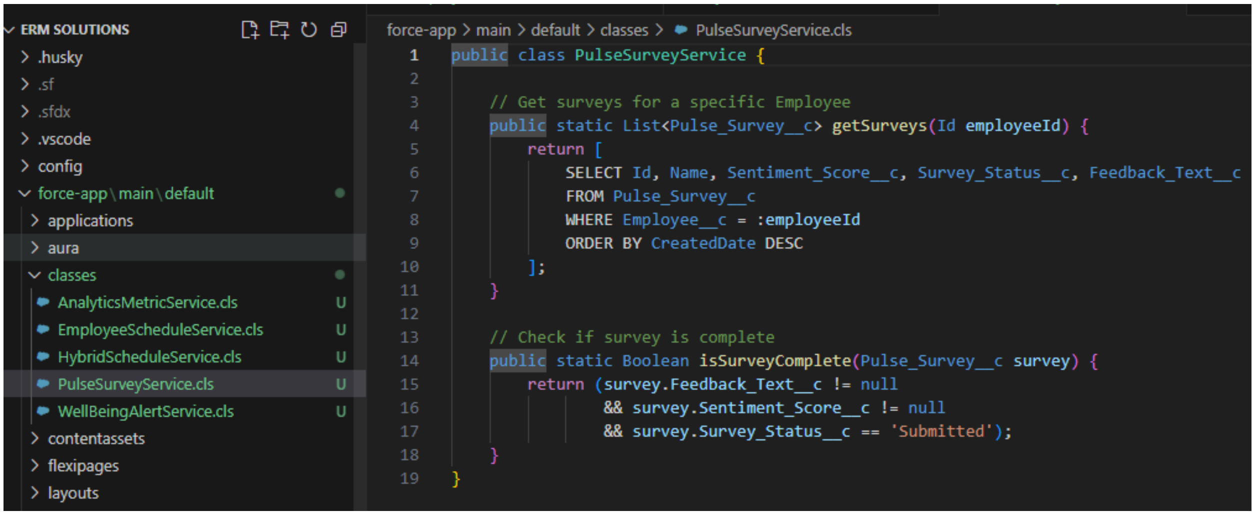
Developed custom LWCs to enhance interactivity:

1. **Hybrid Schedule Viewer LWC** → Displays upcoming and past schedules in a data table.
2. **Well-Being Alert LWC** → Shows critical alerts and allows acknowledgment (with events).
3. **Pulse Survey Form LWC** → Enables employees to submit survey feedback interactively.



```
lwc
  hybridScheduleList.html          U
  hybridScheduleList.js            U
  hybridScheduleViewer.js-meta.xml U
  pulseSurveyForm.html             U
  pulseSurveyForm.js               U
  wellBeingAlert.html              U
  wellBeingAlert.js                U
objects
```

7. Apex with LWC



```
ERM SOLUTIONS
> .husky
> .sf
> .sfdx
> .vscode
> config
v force-app\main\default
  > applications
  > aura
  v classes
      AnalyticsMetricService.cls   U
      EmployeeScheduleService.cls  U
      HybridScheduleService.cls    U
      PulseSurveyService.cls       U
      WellBeingAlertService.cls    U
  > contentassets
  > flexipages
  > layouts
```

force-app > main > default > classes > PulseSurveyService.cls

```
public class PulseSurveyService {

    // Get surveys for a specific Employee
    public static List<Pulse_Survey__c> getSurveys(Id employeeId) {
        return [
            SELECT Id, Name, Sentiment_Score__c, Survey_Status__c, Feedback_Text__c
            FROM Pulse_Survey__c
            WHERE Employee__c = :employeeId
            ORDER BY CreatedDate DESC
        ];
    }

    // Check if survey is complete
    public static Boolean isSurveyComplete(Pulse_Survey__c survey) {
        return (survey.Feedback_Text__c != null
                && survey.Sentiment_Score__c != null
                && survey.Survey_Status__c == 'Submitted');
    }
}
```

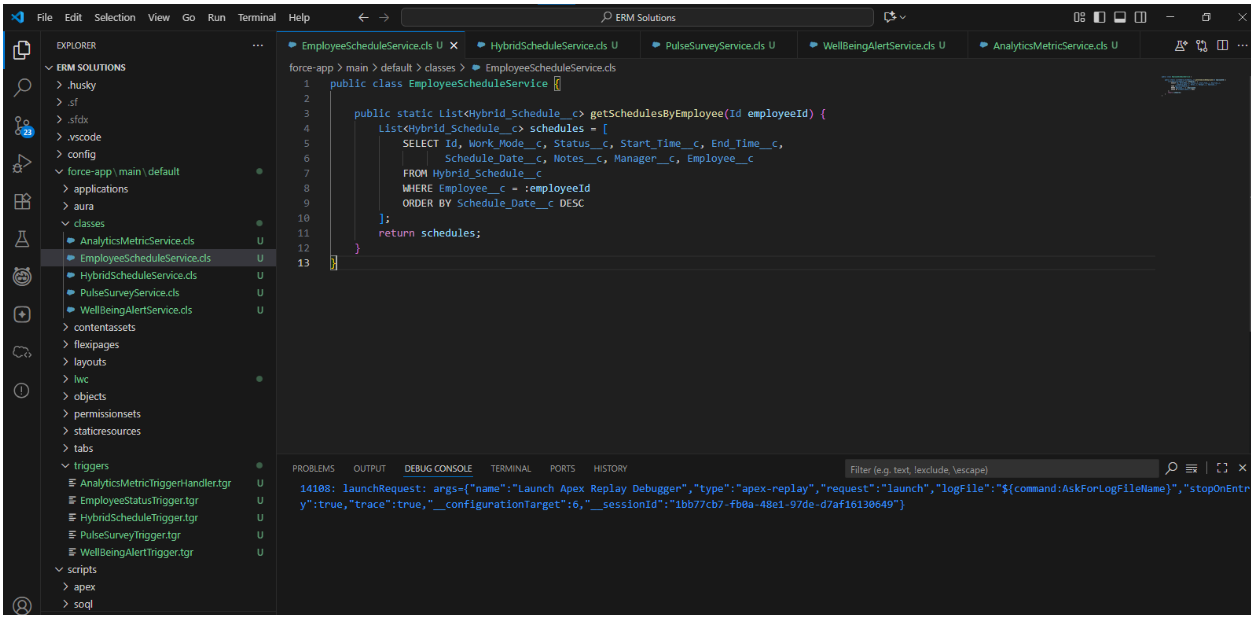
- Created supporting **Apex classes** for each object:
    - `HybridScheduleService.cls` → Fetches schedules.

  - `WellBeingAlertService.cls` → Retrieves critical well-being alerts.
  - `PulseSurveyService.cls` → Submits employee survey feedback.
- Integrated LWCs with these Apex classes using `@AuraEnabled` methods.

---

8. Events in LWC

- Implemented **Custom Events** in LWCs:
  - Example: `alertack` event in Well-Being Alert LWC notifies parent when an alert is acknowledged.
- Enabled **parent-child communication** and **pub-sub messaging patterns** for dashboard updates.


File
Edit
Selection
View
Go
Run
Terminal
Help
ERM Solutions
EXPLORER
ERM SOLUTIONS
.husky
.sf
.sfdx
.vscode
config
force-app \ main \ default
applications
aura
classes
AnalyticsMetricService.cls
EmployeeScheduleService.cls
HybridScheduleService.cls
PulseSurveyService.cls
WellBeingAlertService.cls
contentassets
flexipages
layouts
lwc
objects
permissionsets
staticresources
tabs
triggers
AnalyticsMetricTriggerHandler.tgr
EmployeeStatusTrigger.tgr
HybridScheduleTrigger.tgr
PulseSurveyTrigger.tgr
WellBeingAlertTrigger.tgr
scripts
apex
soql
EmployeeScheduleService.cls U
HybridScheduleService.cls U
PulseSurveyService.cls U
WellBeingAlertService.cls U
AnalyticsMetricService.cls U
force-app > main > default > classes > EmployeeScheduleService.cls
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
HISTORY
Filter (e.g. text, !exclude, \escape)

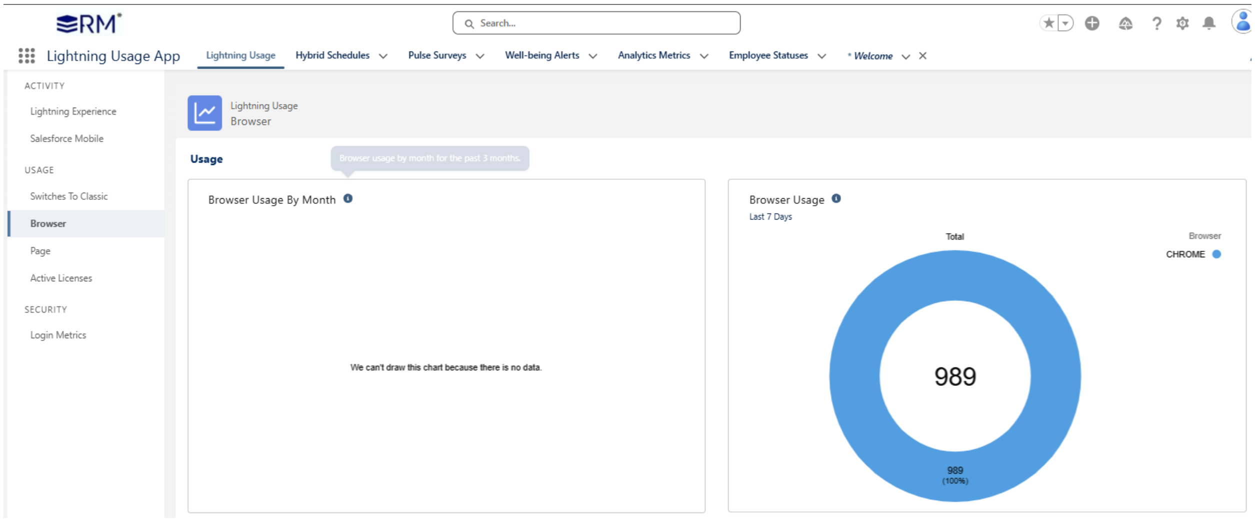
```
public class EmployeeScheduleService {

    public static List<Hybrid_Schedule__c> getSchedulesByEmployee(Id employeeId) {
        List<Hybrid_Schedule__c> schedules = [
            SELECT Id, Work_Mode__c, Status__c, Start_Time__c, End_Time__c,
                   Schedule_Date__c, Notes__c, Manager__c, Employee__c
            FROM Hybrid_Schedule__c
            WHERE Employee__c = :employeeId
            ORDER BY Schedule_Date__c DESC
        ];
        return schedules;
    }
}
```

```
14108: launchRequest: args={"name":"Launch Apex Replay Debugger","type":"apex-replay","request":"launch","logFile":"${command:AskForLogFileName}","stopOnEntr
y":true,"trace":true,"__configurationTarget":6,"__sessionId":"1bb77cb7-fb0a-48e1-97de-d7af16130649"}
```

---

9. Wire Adapters

- Used `@wire` to fetch data directly from Apex classes.
- Example: Wired schedules list from `HybridScheduleService` into the Schedule Viewer LWC.
- Enabled auto-refresh when data changes.



We can't draw this chart because there is no data.

Phase 6 Status: Completed Successfully

Next Steps:

- Proceed to **Phase 7: Integration & External Access**
- Begin testing LWC, Lightning App Builder, Apex with LWC… etc

---

Phase 6 Completion Document.
