**Project Title: Talent Exchange- A Skill Exchange Marketplace.**

**Industry:** Technology Services

**Project Type:** B2B Salesforce Implementation

**Requirement Gathering:**

The reality is that many organizations do want to help their employees develop and learn new skills. But it’s usually difficult for employees to identify someone they can turn to, or seek help from. Managers and HR teams need a solution to help quickly connect people by skill, monitor learning progress and provide guidance — without more coordination or paperwork on their end.

**Stakeholder Analysis:**

Employees want to learn and improve in other areas even but often don’t know where to start or who can help them.

Learning & Development teams are responsible for training but struggle to personalize learning for a large workforce.

HR leaders aim to build trust, reduce anxiety about job security, and create an environment where employees feel supported and empowered.

Each group needs a solution that’s simple, clear, and helps build confidence — while making learning feel like a natural part of their daily work life.

**Business Process Mapping:**

Right now, requests to learn something new are often sent through email or recorded in spreadsheets, making it hard to keep track. Managers have to manually monitor progress, and there’s little way to gather feedback or know if learning is actually helping people improve. This leads to wasted time and missed opportunities to grow.

So, employees can:

Create their skill profiles and learning requests easily on one platform.

Be automatically matched with mentors based on expertise, availability, and interests.

Access dashboards that give managers and HR clear insights into learning trends and skills development.

**Phase 2: Org Setup & Configurator**

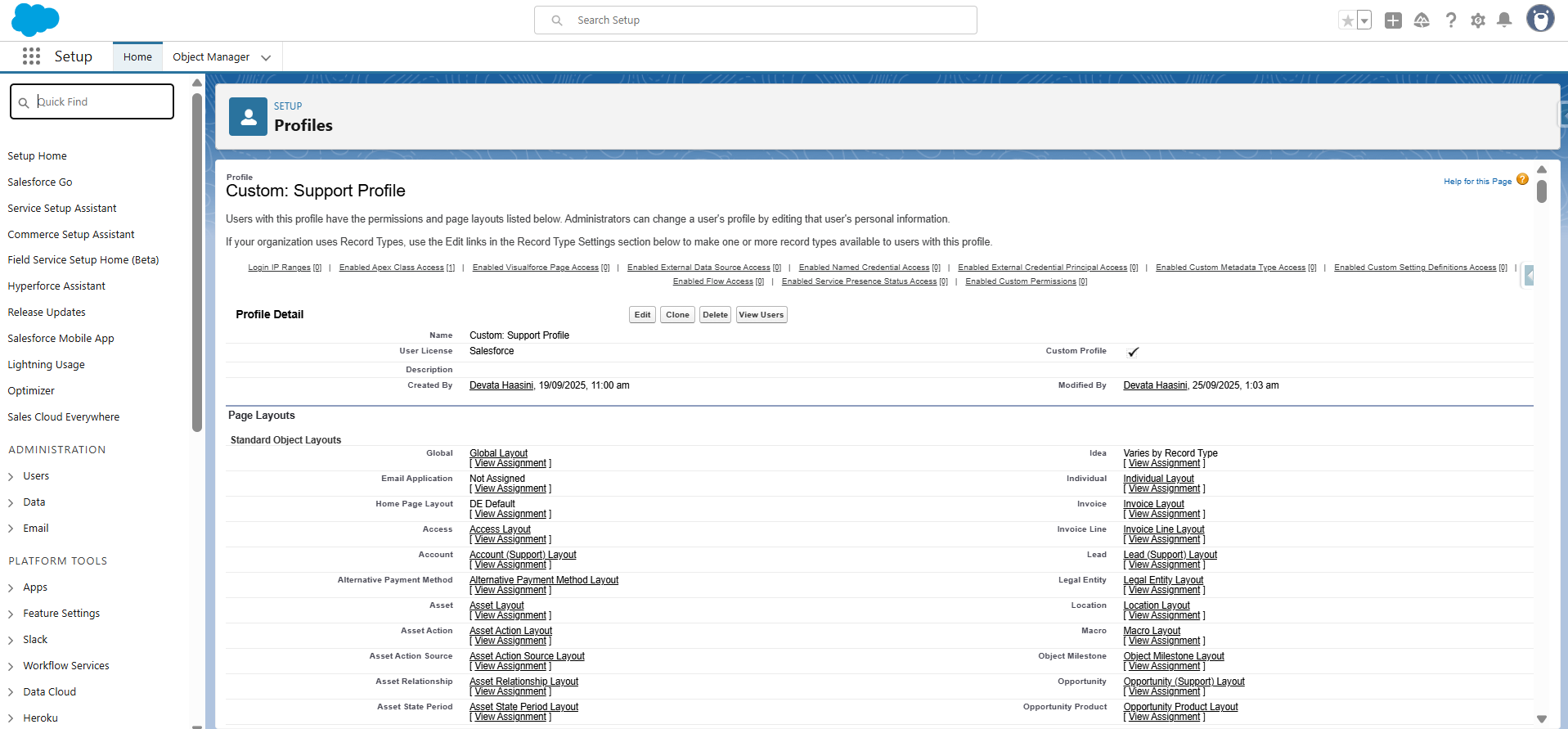
Before creating the skill-sharing platform, the salesforce environment consists of:

**Company Profile Setup:** I configured this to set the basic details of the company, such as its name, currency, and time zone. This ensures all financial data and timestamps in the Talent Exchange system are consistent and accurate.

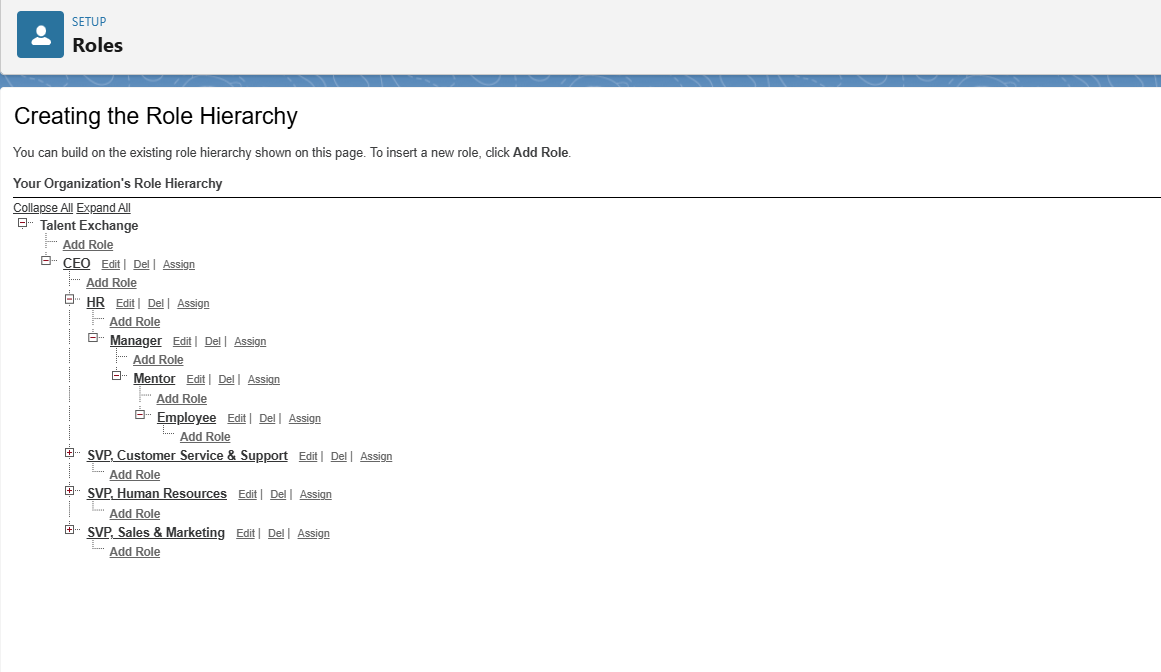
**Business Hours & Holidays**  
 Defined business hours to restrict when sessions could be scheduled and set holiday calendars so mentors/learners were not auto-assigned during non-working days.

**User Setup & Licenses:** I created users for the HR team and for different employees to act as mentors. This was crucial for testing the functionality from various perspectives and for granting each user a Salesforce Platform license to access the custom objects we built.

**Profiles:** I used Profiles to define the baseline permissions for the users. I used a profile to give the HR team access to all fields on the Mentor\_Metrics\_\_c object, while giving standard employees a more limited view.



**Roles:**I set up Roles to create a data hierarchy. This was used to allow managers and HR leaders to see the data for all employees who report to them, a key requirement for the project. The HR Director role could see all HR Manager records.



**Permission Sets:** This was the most important tool for access control. I used Permission Sets to grant specific permissions on top of a user's profile. For instance, a profile might not have access to Apex classes, but you used a permission set to grant access to the MentorPortalController and MentorMetricsBatch Apex classes for the relevant users.

**OWD (Organization-Wide Defaults):** I set a restrictive OWD for custom objects. This ensures that only the record owner or those higher in the role hierarchy can see a specific record unless sharing is explicitly granted. By setting the Session\_\_c object to Private, ensured that each mentor's sessions were not visible to all other users.

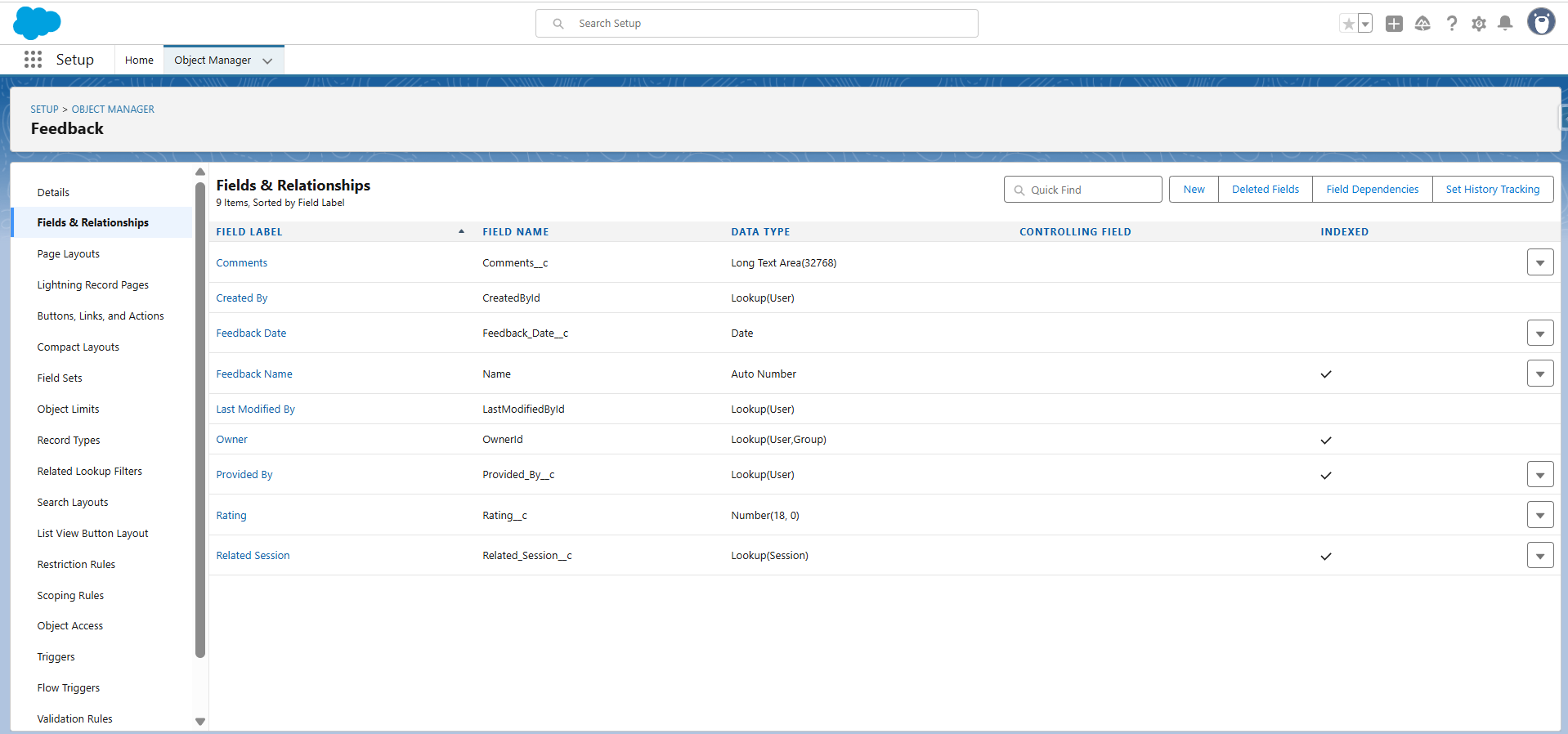
**Sharing Rules:** Sharing Rules are used to open up access for specific groups of users. For example, created a sharing rule to ensure that HR managers could see all mentor metrics, regardless of who owns the records.

**Phase 3: Modelling the Data & Relationships**

In this phase I will structure the data as so that all are linked in an easy-to-navigate way.

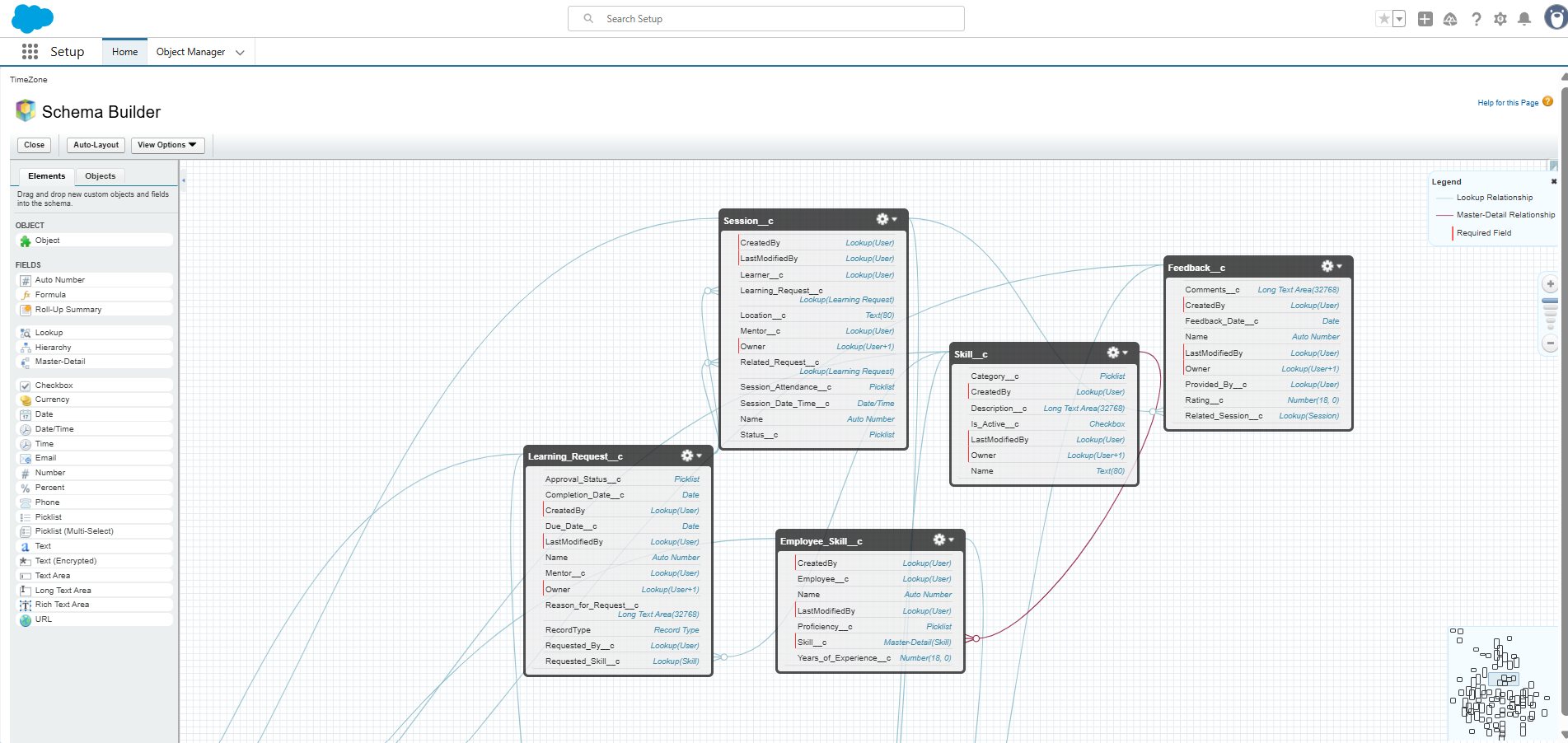
**Standard & Custom Objects**: I used custom objects for this project. Learning\_Request\_\_c to capture skill learning needs from employees. Mentor\_Metric\_\_c to track mentor performance (sessions taught). **Feedback\_\_c** to capture learner feedback after sessions.

**Fields**: Created custom Fields on each of your objects to store specific information.Some of them are *Learning\_Request\_\_c*: Skill Needed, Request Status, Requested By, Assigned Mentor. *Mentor\_Metric\_\_c*: Avg Feedback Rating, Total Sessions, Performance Score.



**Page & Layout Design**: Designed layouts so each user role only sees what matters. Some of them are Learner layout: simple → request skill, see mentor assigned and Mentor layout: session details , feedback fields.

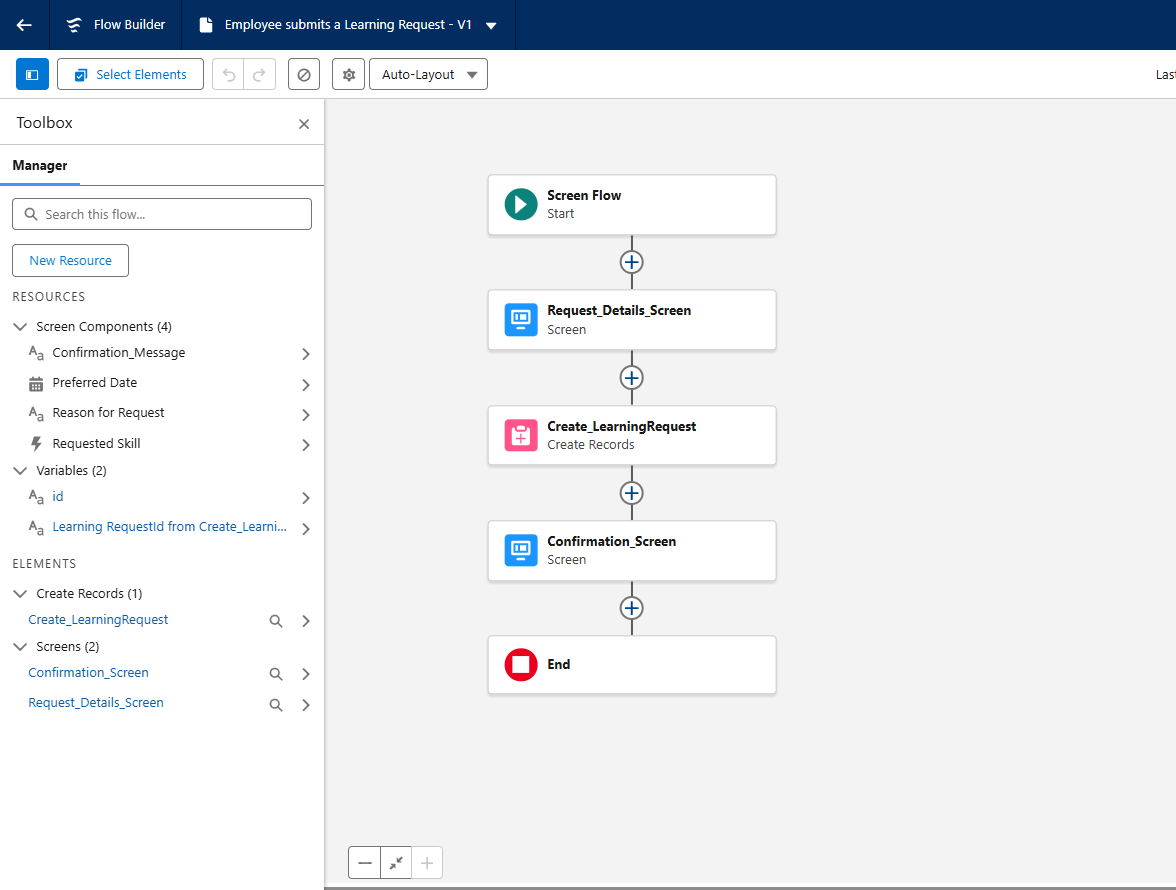
**Schema Mapping**: Structure how things fit together to make it easy to manage and access data.



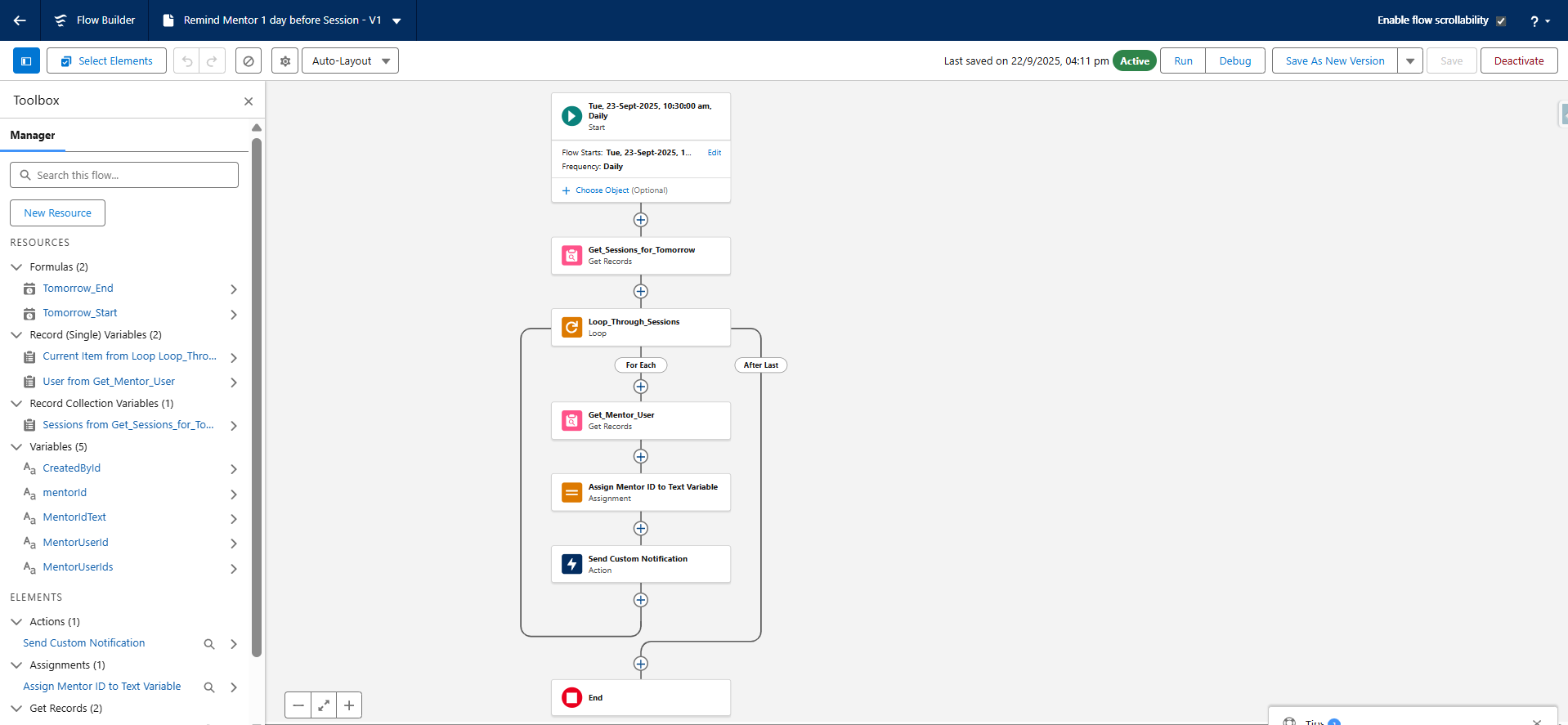
Lookup, Master-Detail & Hierarchical Relationships: Join records of connecting mentors to requests, all while keeping data safe and organized.

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

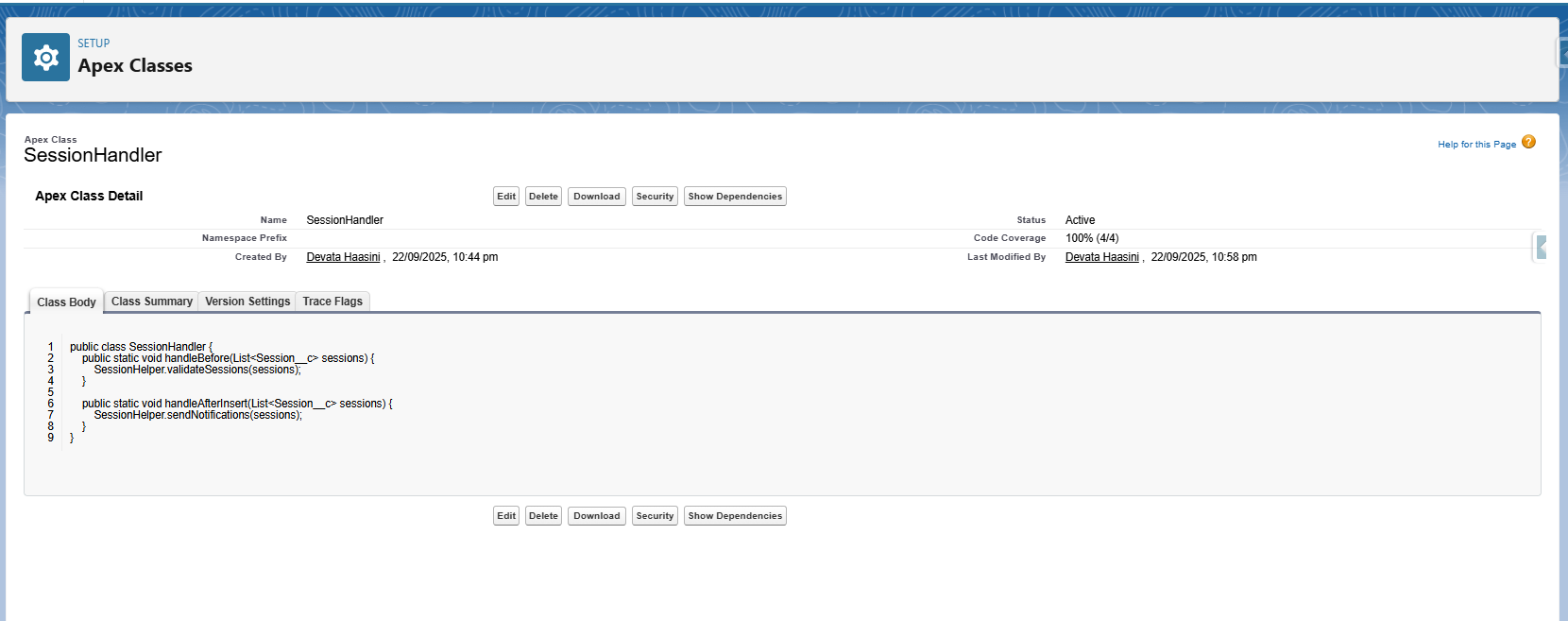
* **Validation Rules:** Have employees input real skills and experience when requesting learning. Used to ensure only valid data is entered.For example, Learner must select at least one skill in a request, Mentor cannot submit feedback without entering a rating.
* **Flow Builder**: This was the central automation tool for your project. Used screen flow and Record-Triggered Flow ,where Record-Triggered Flow to handle a key business process: notifying the HR team when a mentor achieves "Top Mentor" status. I used 4 flows , 1 screen flow and 3 record triggered flows.

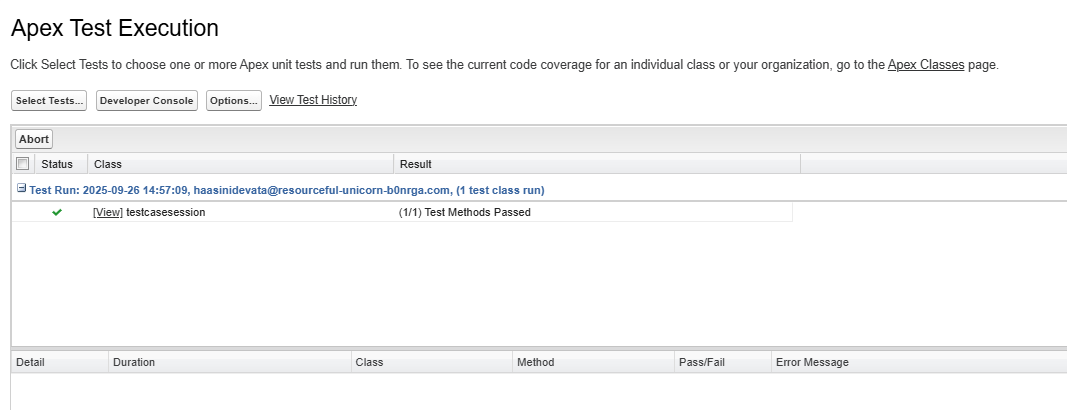
****

* **Email Alerts & Personalized Notifications:** As an action within the Flow, used an Email Alert to send a notification. The alert was configured to dynamically merge data from the Mentor\_Metrics\_\_c record, providing the HR team with an immediate, personalized summary of the top-performing mentor.



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

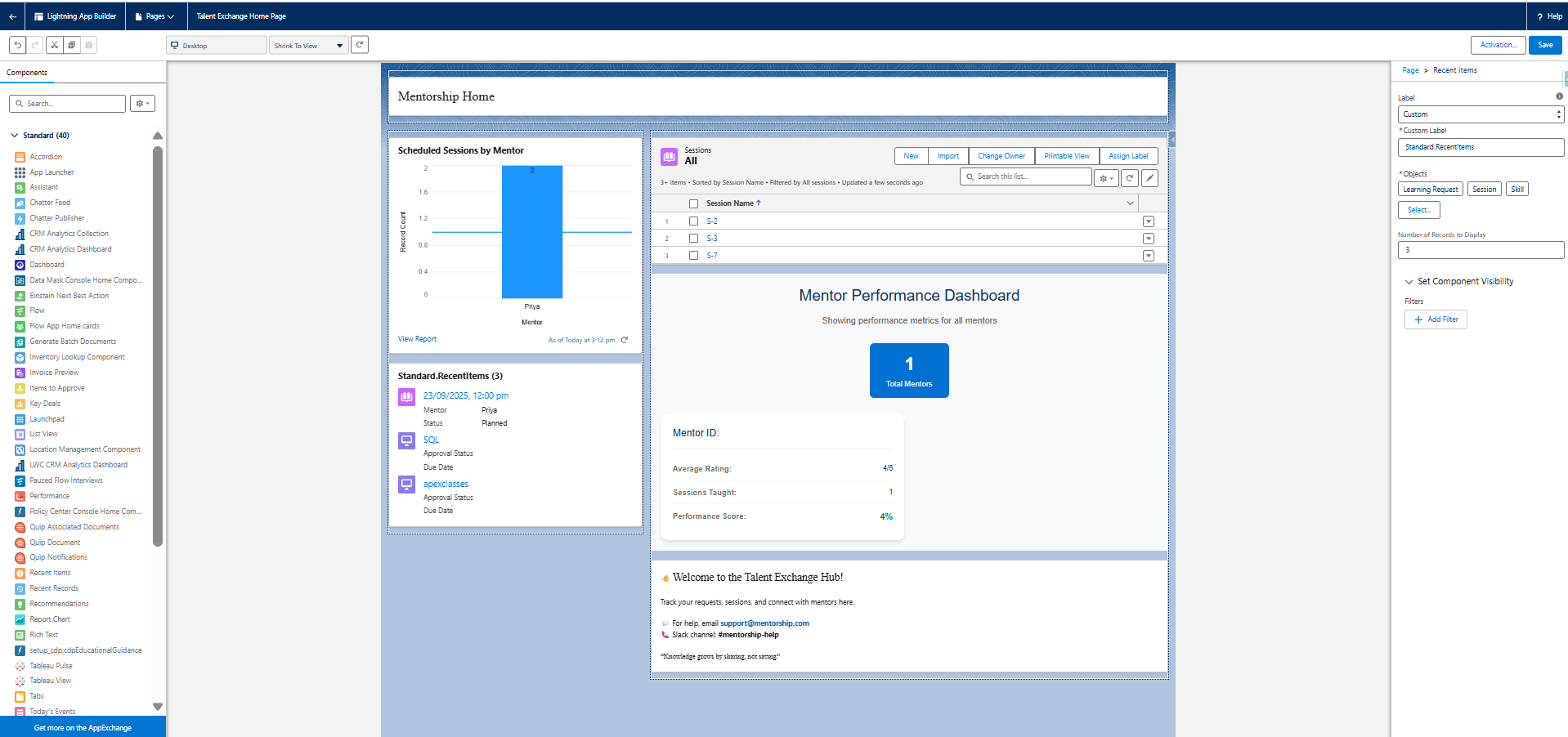
* **Apex Classes**: I used 2 Apex classes for this project.
* LearningRequestHelper - Provides reusable methods for LearningRequestTrigger, like for updating related records, validation logic.
* SessionHelper - Provides reusable methods for SessionHandler, for filtering sessions, calculating attendance counts.
* 
* **Apex Triggers**: Auto update Mentor Match or Send notification on New Learning Request, Feedback Generate.
* LearningRequestHandler - Encapsulates logic for handling Learning Requests like auto-assigning mentors, updating status.
* SessionHandler - Handles logic for Session\_\_c triggers, for updating related Learning Requests, sending notifications.
* **SOQL (Salesforce Object Query Language):** This was a fundamental tool used. The Apex classes I used SOQL to query data from custom objects. (Mentor\_Metrics\_\_c and Session\_\_c). This allowed to retrieve all the necessary data to perform calculations and display it on the mentor dashboard.
* Batch Apex:Automatically calculates metrics from raw Session/Feedback data, Runs in background (scheduled job), Updates Mentor\_Metric\_\_c records automatically
* Scheduled Apex: I used Scheduled Apex to automatically run the MentorMetricsBatch class. This was a core requirement of the project, as it ensured that mentor performance metrics were calculated and updated on a regular, consistent basis, such as every night or every Sunday.
* Test Classes: To Test the code, 2 test cases are used.
  + TestLearningRequest to test LearningRequestHandler and LearningRequestHelper, validates learning requests creation, status updates, and mentor assignments.
  + TestCaseSession to ensure accurate KPIs and bulk-safe processing for dashboards and mentor performance tracking.



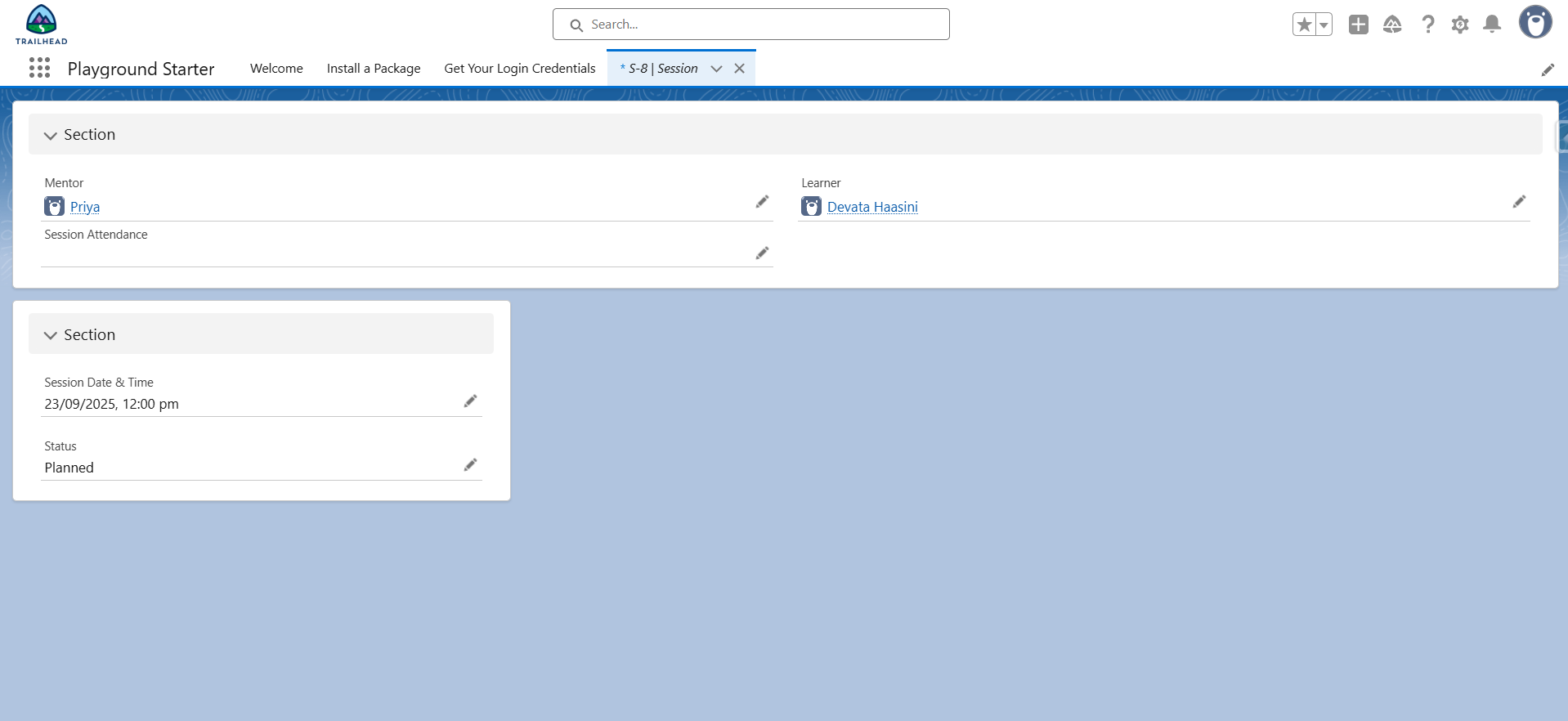
**Phase 6: The development of the user interface**

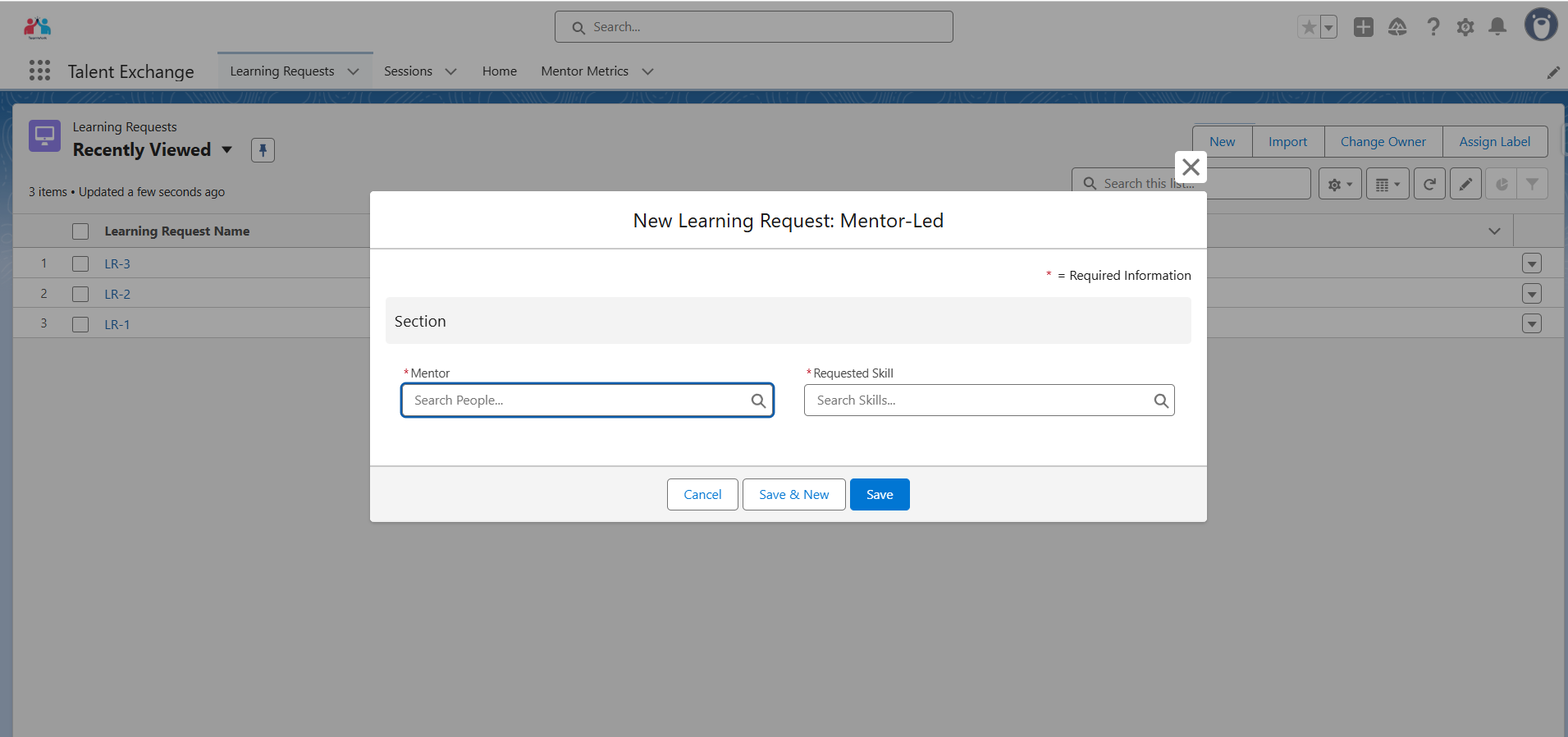
I will make the experience seamless and user-friendly for employees and managers to view skill profiles, submit learning requests and feedback.

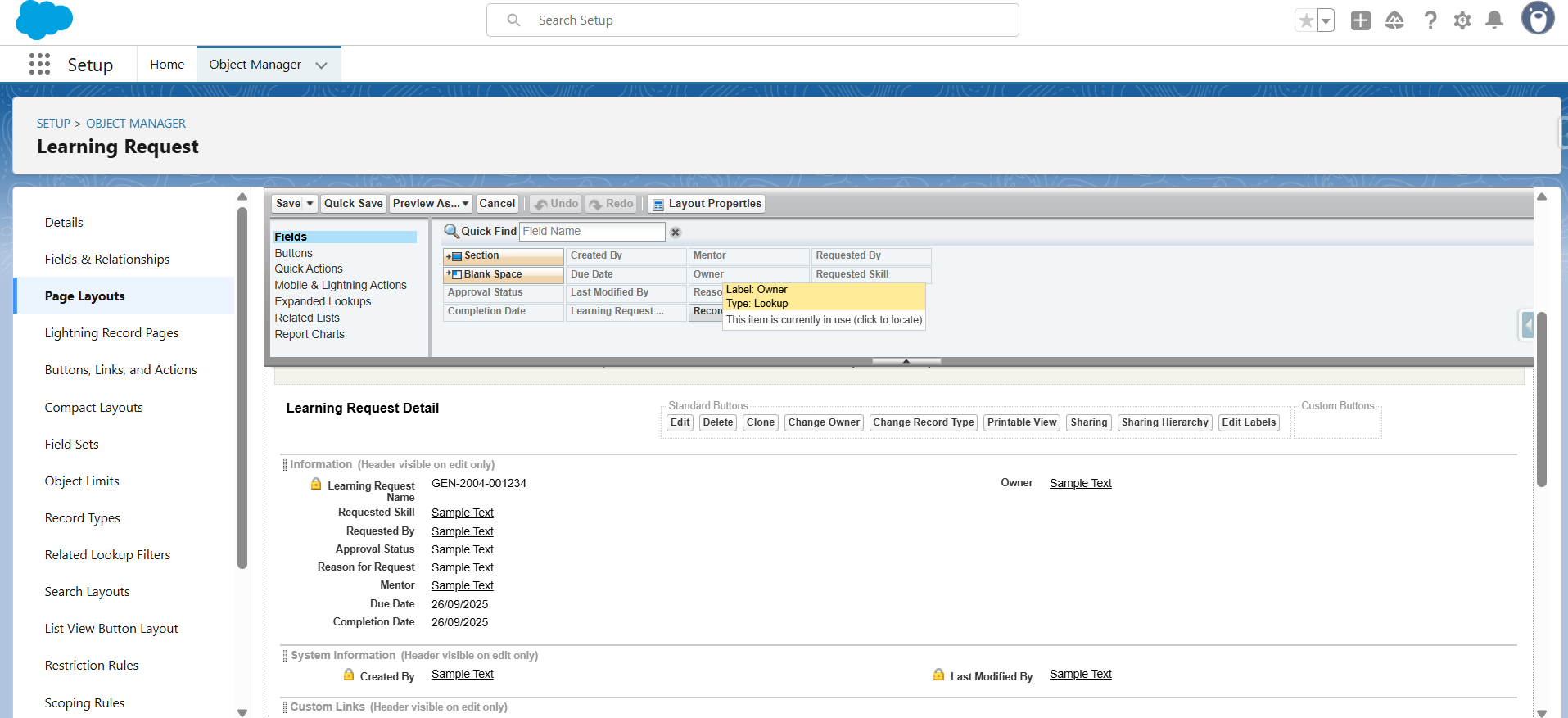
* **Lightning App Builder:** Created custom pages for the Learner Portal.Arranged components like performance summary cards, session tables, and dashboards on the homepage, Simplifies navigation for employees, Managers and HR can view learning insights at a glance.

****

* Tabs : Arrange details in tabs and posting on homepage for instant access.

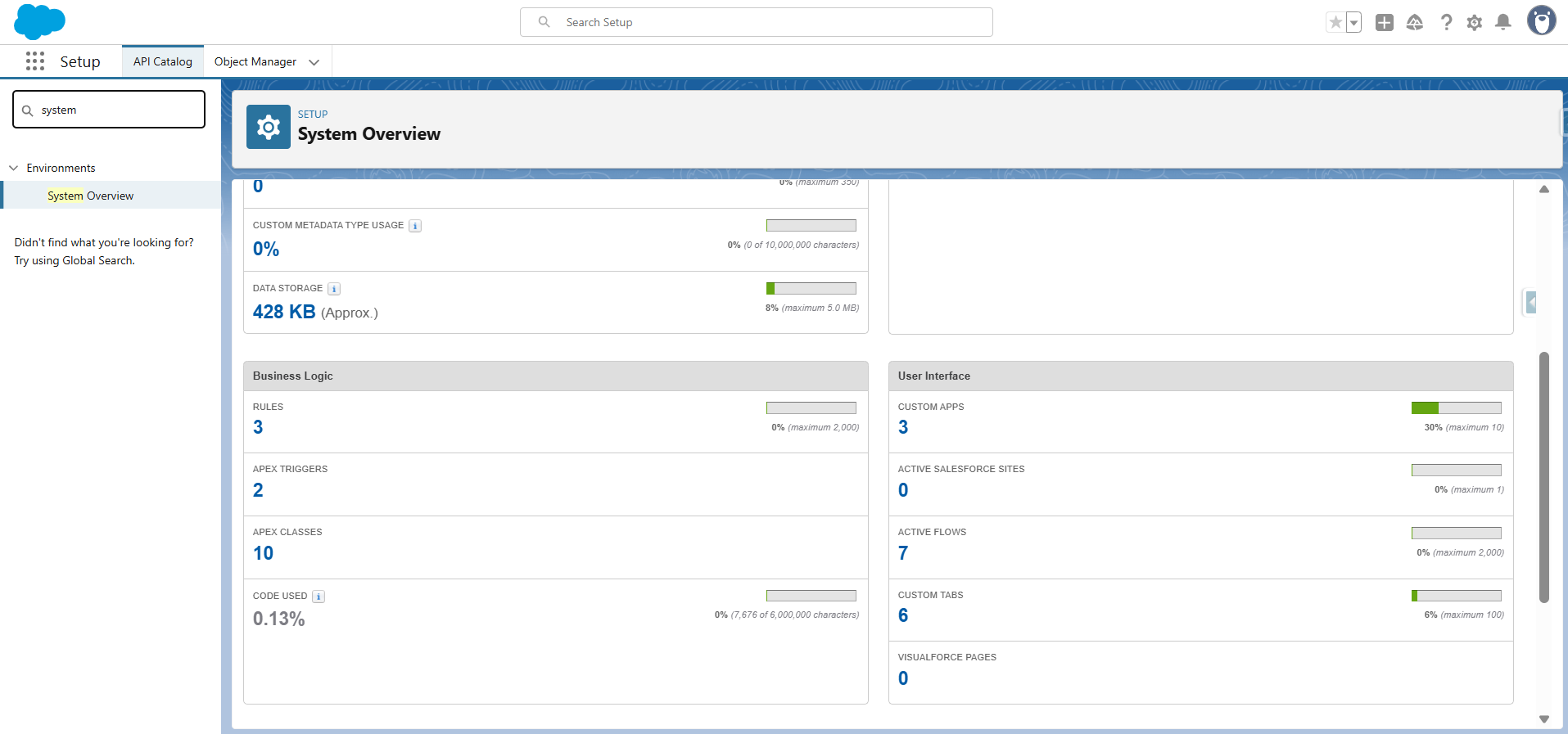




* **LWC (Lightning Web Components):** This is used to create custom visual dashboards that display mentor performance metrics in real-time, transforming raw Salesforce data from the Mentor\_Metric\_\_c object into an interactive, professional-looking interface with cards showing average ratings, session counts, and performance scores, which replaces manual record-checking and provides program managers with immediate insights into mentor effectiveness without needing to run separate reports.
* **Page Layouts:**
* 

**Phase 7: Integration & Outer Access:**

**API limit:** Monitored API usage for batch processing MentorMetricsBatch, fetching aggregated data for LWC dashboards.

****

I didn’t use these Named Credentials, External Services, Web Services (REST/SOAP), Callouts, Platform Events, Change Data Capture, Salesforce Connect, API Limits, OAuth & Authentication, Remote Site Settings as it is not connected externally and didn’t use for real time updates.

**Phase 8: Data Management and Deployment**

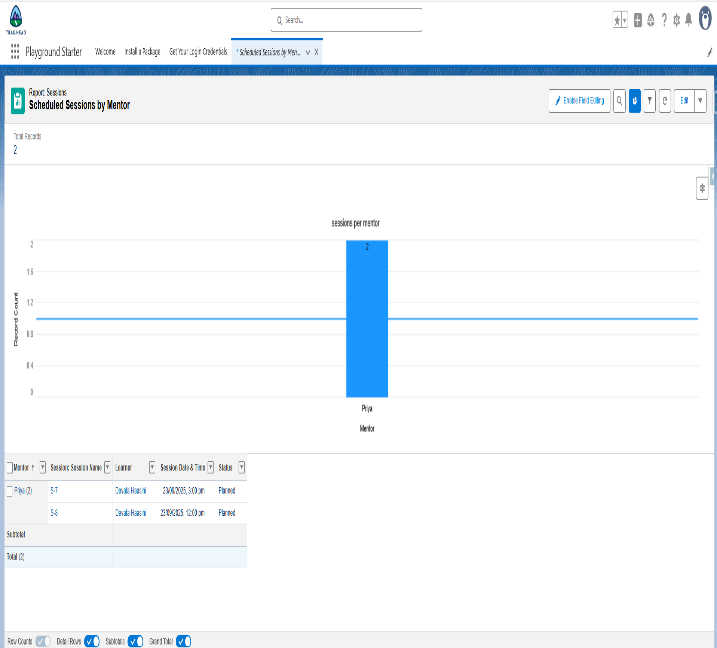
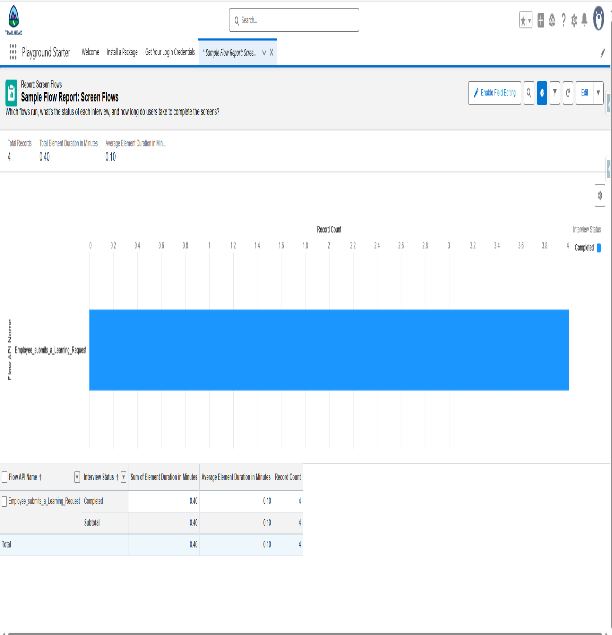
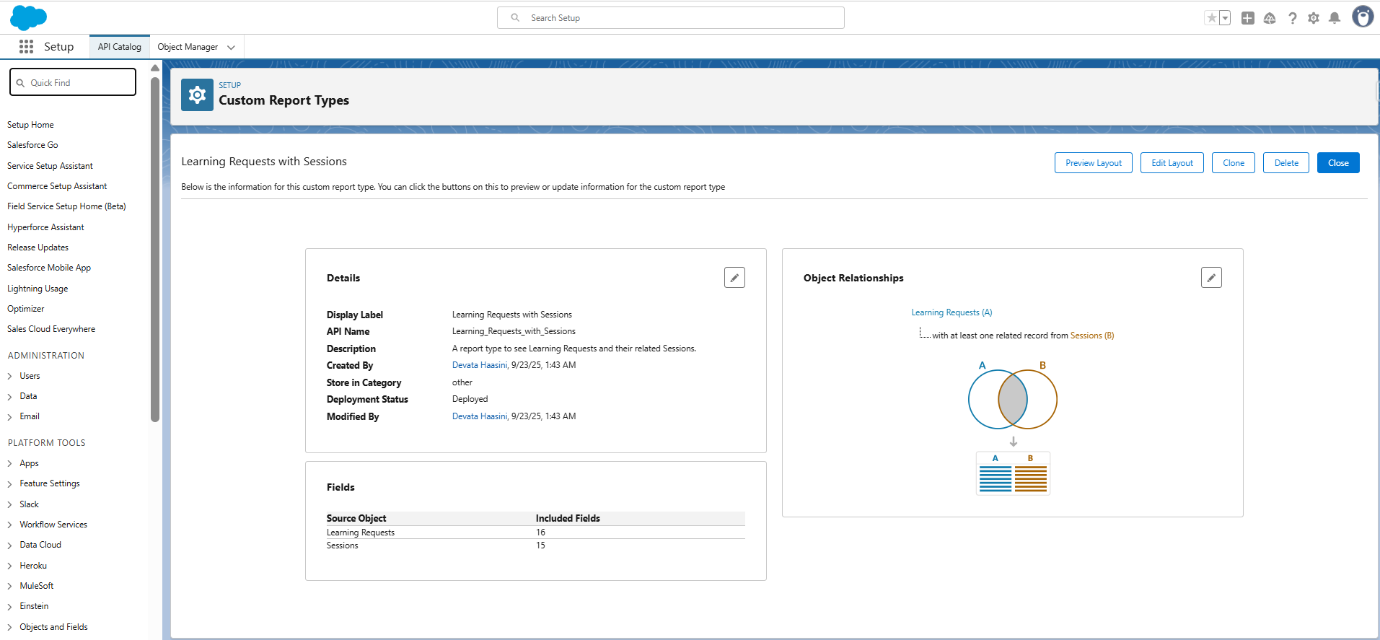
**VS Code & SFDX:** This was the primary tool used for all development and deployment.

* I wrote all of Lightning Web Component locally in VS Code.
* Used the Salesforce DX CLI commands (specifically the sf project deploy start command) to deploy code from the local machine to your Developer Edition org. This provided a modern and efficient way to manage and deploy all your custom code.

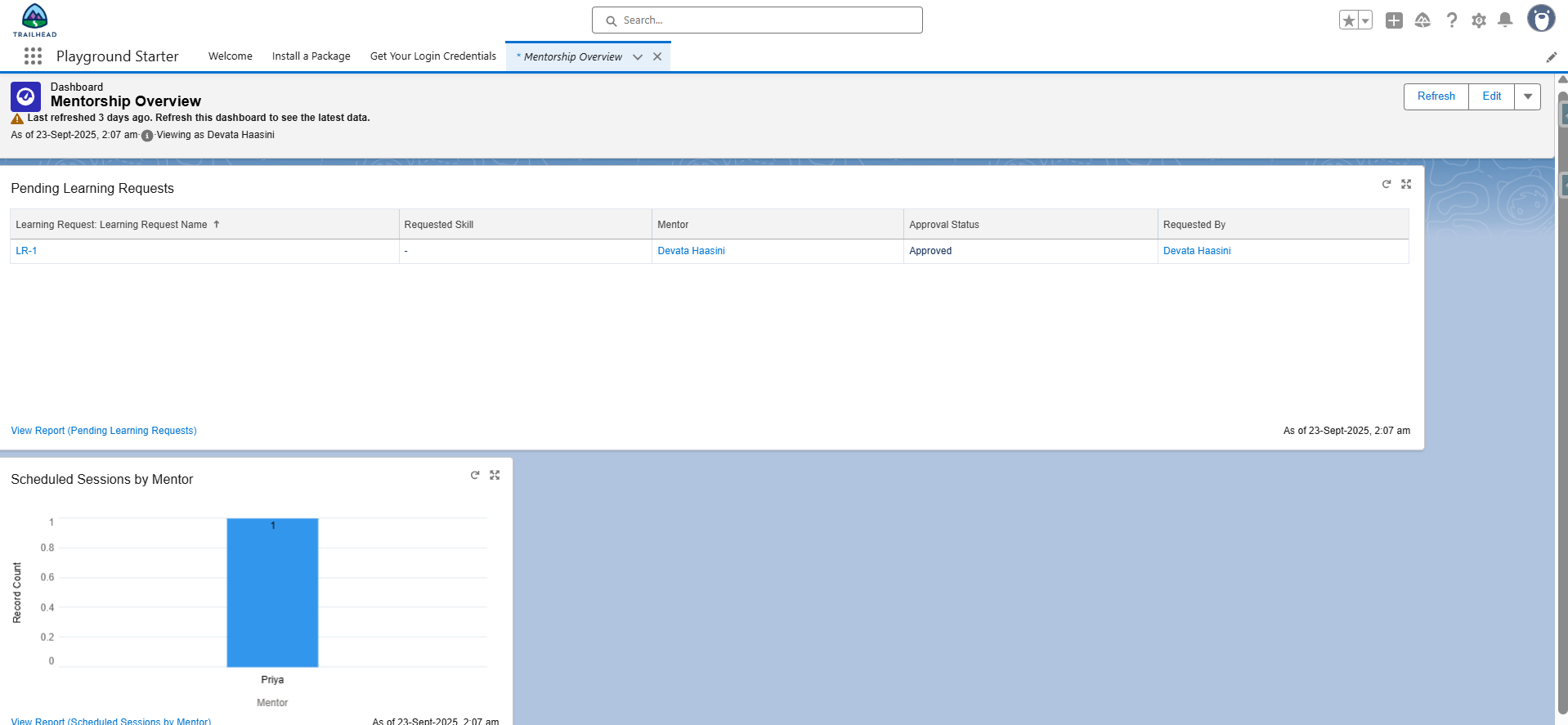
**Phase 9: Reporting, Dashboards & Security Review**

Reports & Report Types: Generate reports to monitor learning status, participation rates and

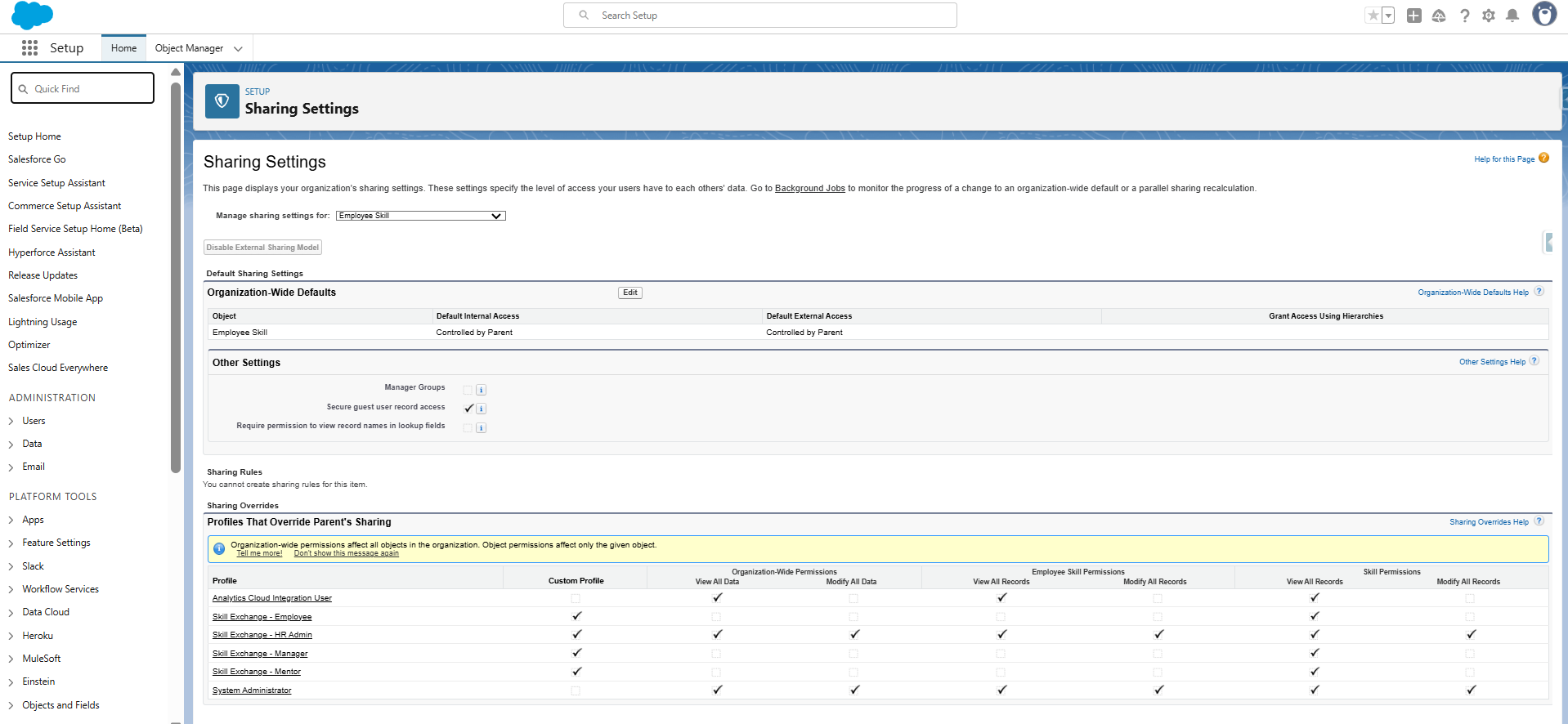
mentor activities.



**Dashboards**: Visual summary for managers to monitor skill development progress.



**Sharing Settings**: Manage who sees what sensitive information.



**Phase 10: Final Presentation & Demo Day**

I wrapped up the project by showcasing how it solves real problems and builds trust across the organization.