

Project Report on
Recruiting Assistance For The HR Managers
Developer - (Short-Term)

Introducion:

Recruiting Assistance For The HR Managers

In this project, we use custom objects, relationships, page layouts to give the HR team easy access to data they need on an existing recruitment app. To make the existing app more efficient for the HR team we create custom objects and relationships to store and access the data more efficiently. We install an unmanaged package in the org to get metadata that acts as existing data in the recruitment app.

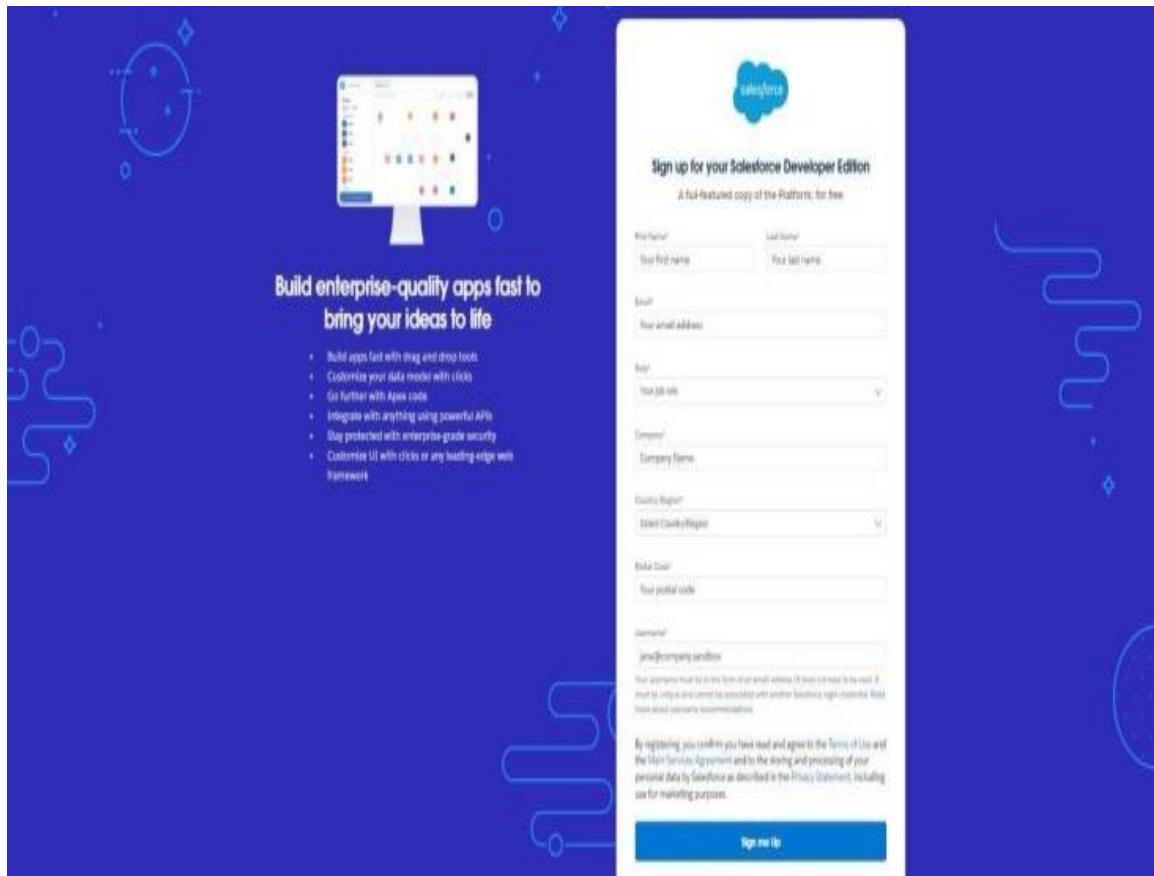
Salesforce

Salesforce is your customer success platform, designed to help you sell, service, market, analyze, and connect with your customers. Salesforce has everything you need to run your business from anywhere. Using standard products and features, you can manage relationships with prospects and customers, collaborate and engage with employees and partners, and store your data securely in the cloud. So what does that really mean? Well, before Salesforce, your contacts, emails, follow-up tasks, and prospective deals might have been organized something like this: <https://youtu.be/r9EX3lGde5k>. There are 5 types of salesforce editions,

1. Essentials: Designed for small businesses getting started with CRM to boost sales or service productivity. It includes a setup assistant and administration tools to customize your deployment as you grow.
2. Professional: Designed for businesses requiring full-featured CRM functionality. It includes straightforward and easy-to-use customization, integration, and administration tools to facilitate any small to midsize deployment.
3. Enterprise: Meets the needs of large and complex businesses. It gives you advanced customization and administration tools, in addition to all the functionality available in Professional Edition, that can support large-scale deployments. Enterprise Edition also includes access to Salesforce APIs, so you can easily integrate with back-office systems.
4. Unlimited: Maximizes your success and extends it across the entire enterprise through the Lightning Platform. It gives you new levels of platform flexibility for managing and sharing all your information on demand. Includes all Enterprise Edition functionality, Premier Support, full mobile access, unlimited custom apps, increased storage limits, and other features.
5. Developer: Provides access to the Lightning Platform and APIs. It lets developers extend Salesforce, integrate with other applications, and develop new tools and applications. Developer Edition also provides access to many of the features available in Enterprise Edition.

Milestone-1

A Developer Org Has All The Features And Licenses You Need To Get Started With Salesforce



1. Search [Developer.salesforce.com](https://developer.salesforce.com).
2. Enter the following details like First name, last name, Email, Role, Company, Country/Region, Postal code, and Username must be unique.
3. Click sign me up, After a few min you will reserve a mail salesforce org and by using the verify account link you can create your new password.
4. Click save.
5. Search login.salesforce.com.
6. By using username and password you can into the salesforce org.
7. By using username and password you can into the salesforce org.
8. The setup page will appear as below.
9. Create a developer org and login with your login credentials.

The screenshot shows a web browser window with multiple tabs open at the top. The active tab is web.whatsapp.com. The main content area displays the Salesforce Developer Edition sign-up page. It features a large blue background image of a computer monitor displaying a network diagram. Below the image, the text "Build enterprise-quality apps fast to bring your ideas to life" is displayed. To the right, there is a form with fields for First Name*, Last Name*, Email*, Role*, and Company*. A "Sign up" button is located to the right of the form. The status bar at the bottom shows several file icons and the date/time: 5:25 PM 10/31/2023.

The screenshot shows a web browser window with multiple tabs open at the top. The active tab is web.whatsapp.com. The main content area displays a Salesforce account verification page. It features a central message "Thanks for signing up with Salesforce!" with a screenshot of the Salesforce interface. Below this, there is a "Verify Account" button and a URL: <https://universitycollegefrntt1.dev-ed.develop.org/salesforce>. The status bar at the bottom shows several file icons and the date/time: 5:25 PM 10/31/2023.

Milestone-2

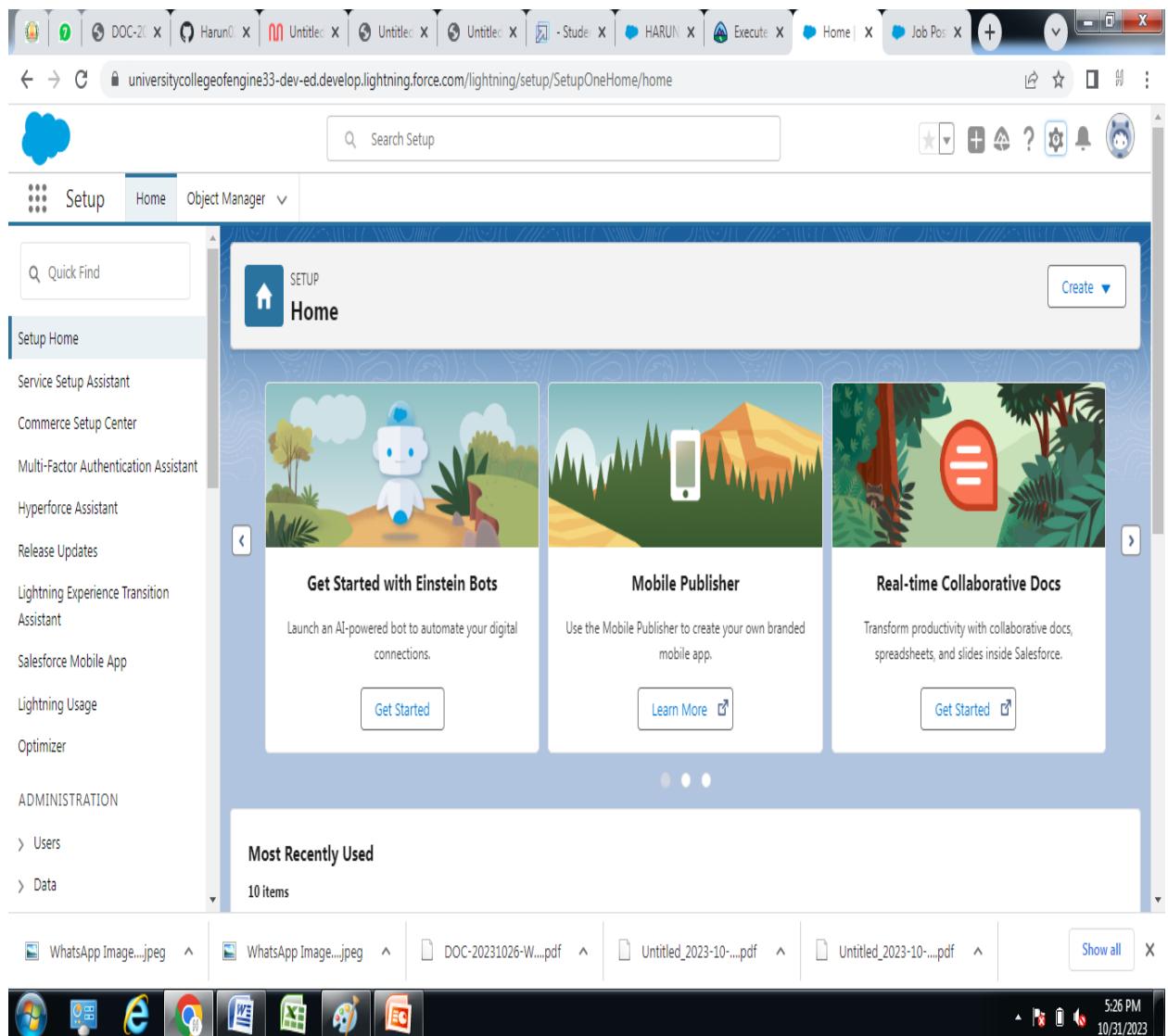
Package Installation

Package installation for Recruiting App

In Salesforce, a package is a collection of Apex classes, triggers, Visualforce pages, and other components that can be installed into an organization. There are two types of packages: managed and unmanaged. Managed packages are developed and distributed by ISVs (Independent Software Vendors) and can be installed from the Salesforce AppExchange, while unmanaged packages are created and distributed by Salesforce administrators within an organization. To install a package, an administrator can navigate to the AppExchange, find the desired package, and click the "Install" button. The administrator will then be prompted to log in to their Salesforce organization and provide permission to install the package.

Click  to launch the App Launcher, then click **Playground Starter** and follow the steps,

1. Click the install a package tab.
2. Paste 04t0P000000N9rs into the field.
3. Click install.
4. Select install for admins only .



Milestone-3

Object

Salesforce objects are database tables that permit you to store data that is specific to an organization. It consists of fields (columns) and records (rows).

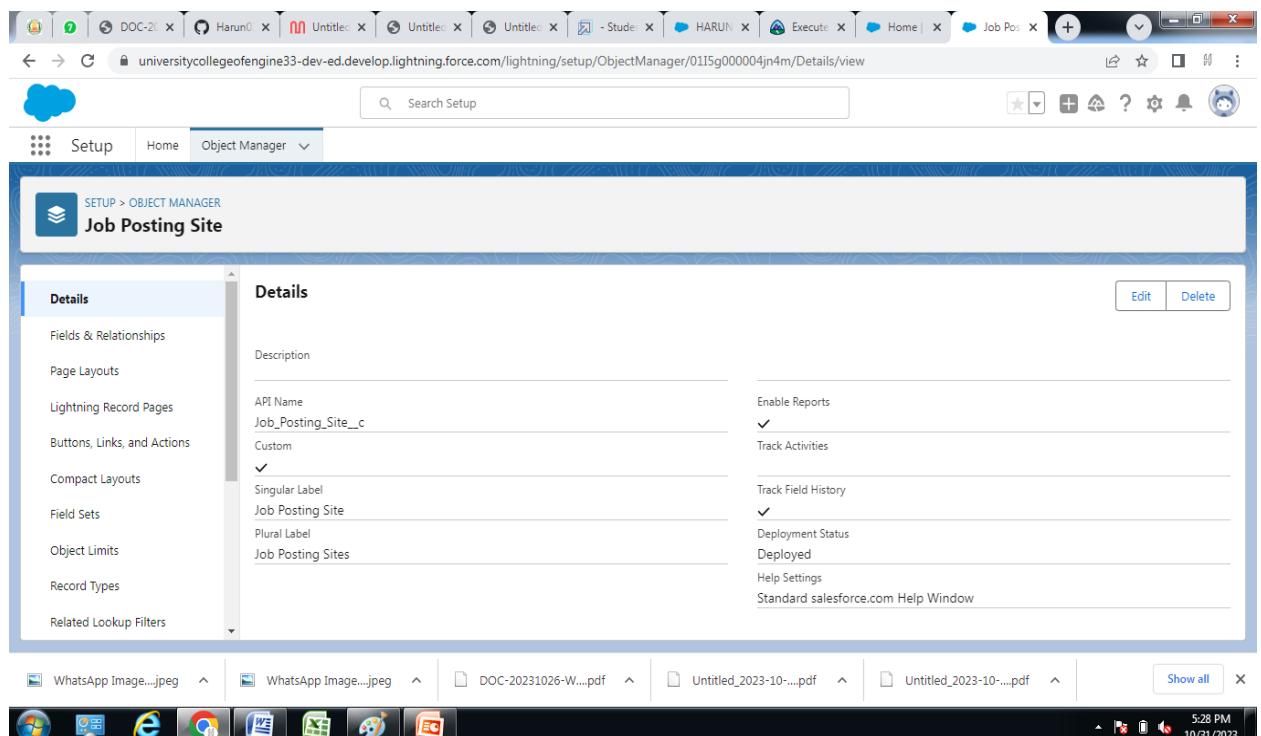
Salesforce objects are of two types:

- Standard Objects: Standard objects are the kind of objects that are provided by salesforce.com such as users, contracts, reports, dashboards, etc.
- Custom Objects: Custom objects are those objects that are created by users. They supply information that is unique and essential to their organization. They are the heart of any application and provide a structure for sharing data.

Create a custom object for Job Posting Sites

To create a custom object, follow these steps :

- Click create, select custom object.
- Fill in the plural label as "**Job Posting Sites**".
- Record name : "**Site Name**"
- Select the data type as "**Text**".
- In the Optional Features section, select Allow Reports and Track Field History.
- In the Deployment Status section, ensure Deployed is selected.
- In the Search Status section, select **Allow Search**.
- In the Object Creation Options section, select select these options:
- Add Notes and Attachments related list to default page layout.
- Launch New Custom Tab Wizard after saving this custom object.
- Leave everything else as is, and click **Save**.

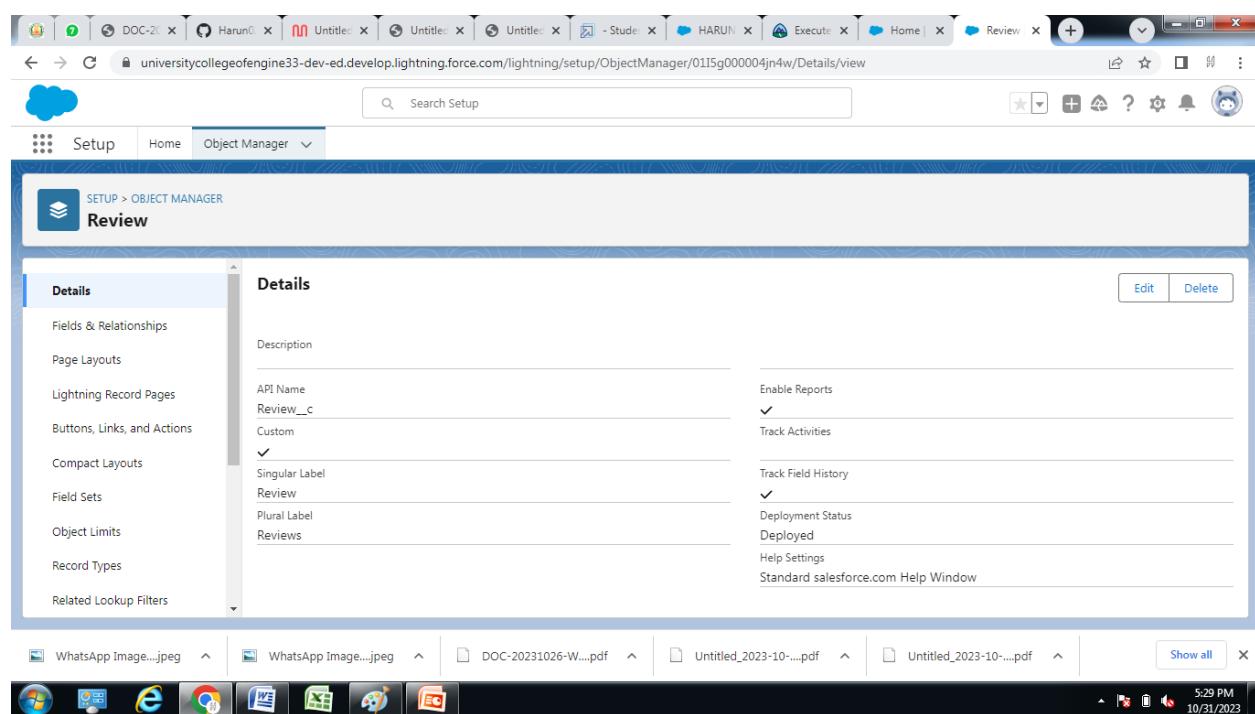


Activity 2

Create a custom object for reviews

To create a custom object, follow these steps :

1. From setup click on object manager.
2. Click create, select custom object.
3. Fill in the label as "Review".
4. Fill in the plural label as "Reviews".
5. Record name : "Review Number"
6. Select the data type as "Auto Number".
7. Under display format enter "REV-{0000}".
8. Enter the starting number as 1.
9. In the Optional Features section, select Allow Reports and Track Field History.
10. In the Deployment Status section, ensure Deployed is selected.
11. In the Search Status section, select Allow Search.
12. In the Object Creation Options section, select Add Notes and Attachments related list to default page layout.
13. Leave everything else as is, and click Save.



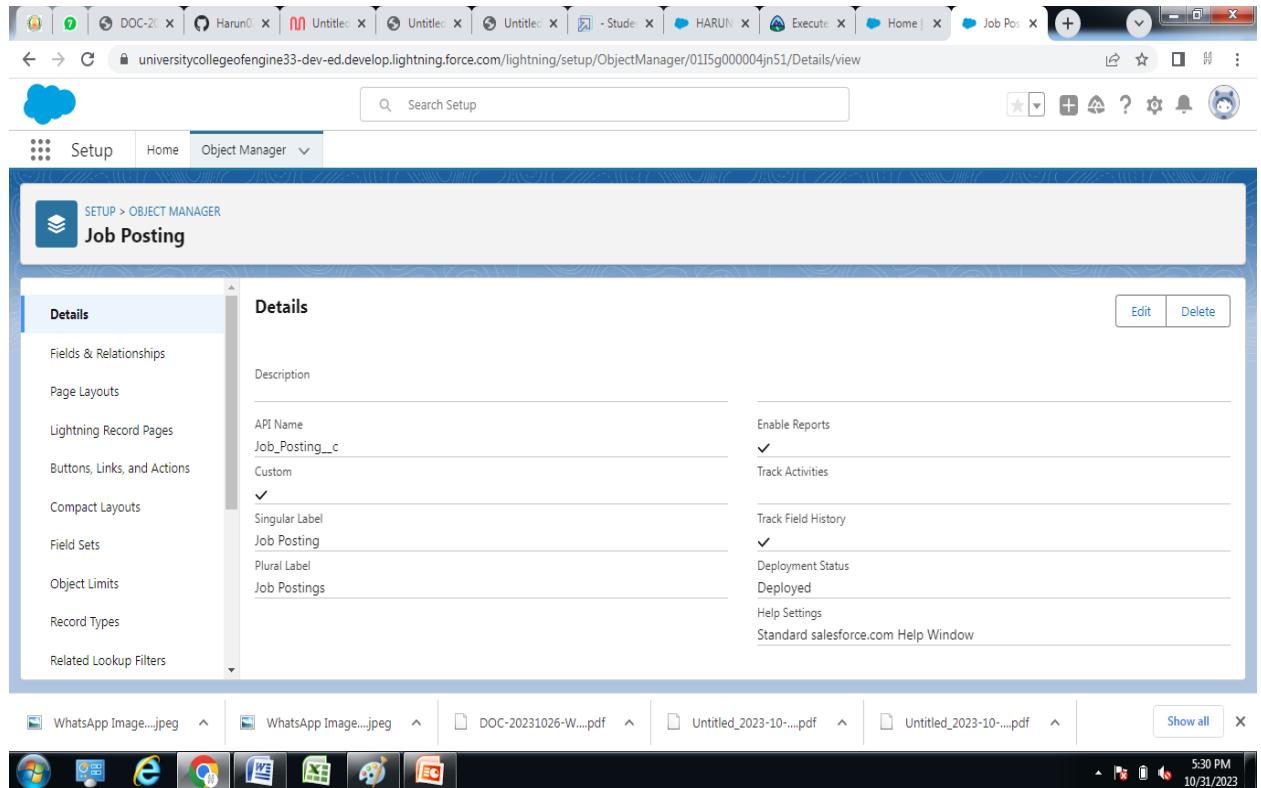
Create A Junction Object For Job Posting

Create a Junction object for Job Posting

To create a custom object, follow these steps :

1. From setup click on object manager.
2. Click create, select custom object.
3. Fill in the label as "Job Posting".
4. Fill in the plural label as " Job Postings".
5. Record name : " Job posting number ".

6. Select the data type as "Auto Number".
7. Under display format enter " JOBPOST-{0000}"
8. Enter the starting number as 1.
9. In the Optional Features section, select Allow Reports and Track Field History.
10. In the Deployment Status section, ensure Deployed is selected.
11. In the Search Status section, select Allow Search.
12. In the Object Creation Options section, select Add Notes and Attachments related list to default page layout.
13. Leave everything else as is, and click Save.



Milestone-4

Tabs

In Salesforce, a tab is a user interface element that allows users to navigate to different sections of the platform, such as Accounts, Contacts, Leads, and Opportunities. Tabs can also be used to access custom objects and custom pages. They are typically located at the top of the screen and can be customized to fit the needs of the organization. There are mainly 4 types of tabs:

1. Standard Object Tabs:

Standard object tabs display data related to standard objects.

2. Custom Object Tabs:

Custom object tabs display data related to custom objects. These tabs look and function just like standard tabs.

3. Web Tabs:

Web Tabs display any external Web-based application or Web page in a Salesforce tab.

4. Visualforce Tabs:

Visualforce Tabs display data from a Visualforce Page.

Now create a custom tab

1. Click on Home tab, enter Tabs in Quick Find and select Tabs.
2. Leave the profile as is and click Next.
3. In the Add to Custom Apps section:
4. Deselect Include Tab.
5. Select Append tab to users' existing personal customizations.
6. Click Save.

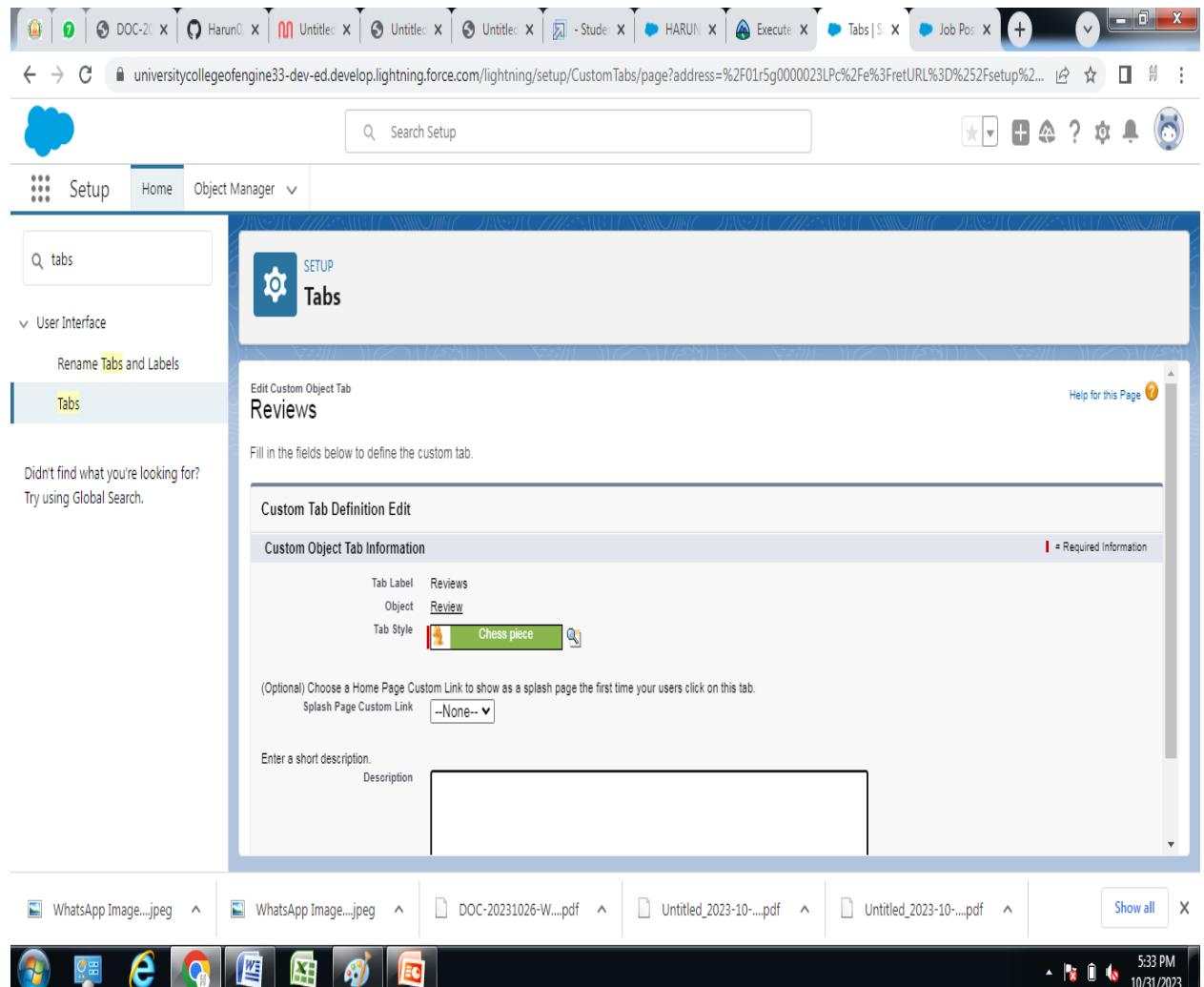
The screenshot shows the Salesforce Setup Home page. The top navigation bar includes tabs for Setup, Home, and Object Manager. A search bar at the top right contains the text "tabs". On the left, a sidebar under "User Interface" has "Tabs" selected. The main content area is titled "SETUP Home" and features three cards: "Get Started with Einstein Bots", "Mobile Publisher", and "Real-time Collaborative Docs". Below these cards is a section titled "Most Recently Used" which lists several recent documents. At the bottom of the page is a footer with a toolbar and system status information.

This screenshot shows the "Custom Tabs" section of the Salesforce Setup page. The title is "SETUP Tabs". It includes a brief description of what custom tabs are and how they can be used. Below this is a table for "Custom Object Tabs" with three entries: "Job Postings" (Tab Style: Balls), "Job Posting_Sites" (Tab Style: Car), and "Reviews" (Tab Style: Chess piece). There is also a "Web Tabs" section which currently displays "No Web Tabs have been defined". The interface is identical to the previous screenshot, with the "Tabs" section selected in the sidebar.

Creation Of Reviews Tab

Now create a custom tab

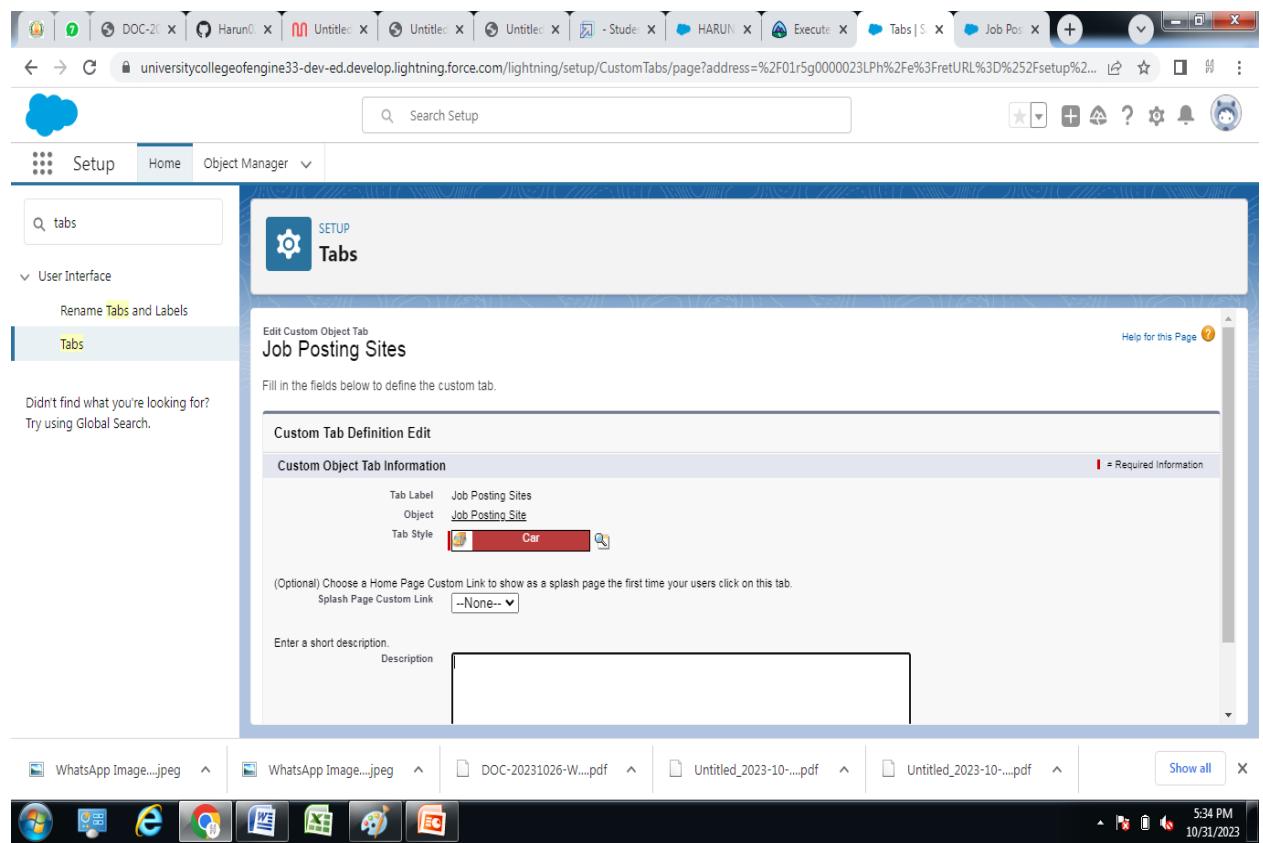
1. Click on Home tab, enter Tabs in Quick Find and select Tabs.
2. Under custom object tabs, click New.
3. For Object, select Reviews.
4. For Tab Style, select any icon.
5. Leave all defaults as is. Click Next, Next, and Save.



Creation Of Job Postings Tab

Now create a custom tab

1. Click on Home tab, enter Tabs in Quick Find and select Tabs.
2. Under custom object tabs, click New.
3. For Object, select Job Postings.
4. For Tab Style, select any icon.
5. Leave all defaults as is. Click Next, Next, and Save.



Milestone-5

Lightning App

Apps in Salesforce are a group of tabs that help the application function by working together as a unit. It has a name, a logo, and a particular set of tabs. The simplest app usually has just two tabs.

There are 2 types of Salesforce applications:

- Standard apps: these apps come with every occurrence of Salesforce as default. Community, Call Center, Content, Sales, Marketing, Salesforce Chatter, Site.com, and App Launcher are included in these apps. The description, logo, and label of a standard app cannot be altered.
- Custom apps: these apps are created according to the needs of a company. They can be made by putting custom and standard tabs together. Logos for custom apps can be changed.

Activity 1

1. Click to launch the App Launcher, then click Recruiting and follow the steps.
2. Click the pencil icon at the top right of the screen.
3. Click Add more Items.
4. From the menu on the left, click All.
5. Next to Job Posting Sites, click the +.
6. Click Add 1 Nav item.
7. Click save.

Milestone-6

Fields And Relationship

Fields in Salesforce represents what the columns represent in relational databases. It can store data values which are required for a particular object in a record. There are 2 types of fields in salesforce:

- Standard fields: There are four standard fields in every custom object that are Created By, Last Modified By, Owner, and the field created at the time of the creation of an object. These fields cannot be deleted or edited and they are always required. For standard objects, the fields which are present by default in them and cannot be deleted from standard objects are standard fields.
- Custom fields: The Custom fields which are added by the administrator/developer to meet the business requirements of any organization. They may or may not be required.

Create New Field For Job Posting Site

From the object manager, click on the job posting site, then click on Fields & Relationships.

1. Click the gear icon and select Setup. This launches Setup in a new tab.
2. Click the Object Manager tab next to Home.
3. Select the job posting sites.
4. Click field and relationship than click new.
5. Select the data type as URL.
6. Click next.
7. For Field Label, enter the Job Posting Site URL.
8. Click Next, Next, and click Save & New.

Now let's create the other fields and we must choose the data types of the fields carefully

1. Status(PickList)(Active Inactive)It should be in separate line
2. Select the Checkbox as the Data Type, then click Next
3. For Field Label, enter Technical Site.
4. Select the Text Area as the Data Type, then click Next.

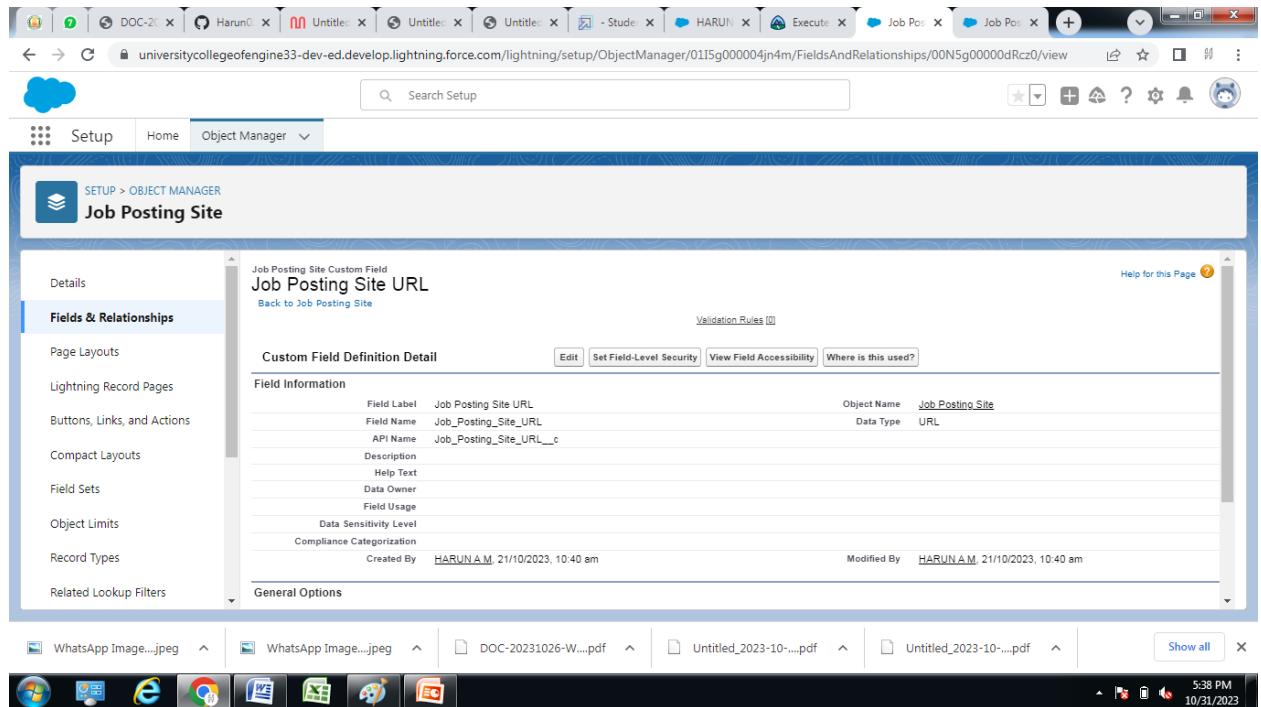
For Field Label, enter Description.

Click Next, Next, then Save & New.

Create pick list field for Job Posting Site object.

Click on the gear icon and then select Setup.

- After the above steps, Select Job Posting Site Object.
- Now Select Fields and relationships from setup menu of the Attendee object.
- Click new and select Pick list fields next and enter label name(Status) and select enter values option(Active, Inactive),next, next and Save.



Create Relationships For Job Posting

Creating a master-detail relationship between Job posting and job posting site.

1. From Setup, go to Object Manager.
 2. Select job posting and click Fields & Relationships.
 3. Click New.
 4. Choose Master-detail Relationship and click Next.
 5. Choose the related Object (Job posting site) and select that object.
 6. Enter the label name(Job Posting Site) for the lookup field.
 7. Click Next, Next, and Save.
- Creating a master-detail relationship between job posting and position for job posting object.
1. From setup, click object manager.
 2. Select Job posting object, click on field and relationships, click new.
 3. Select the data type as Master-detail relationship.

4.Click Next, relate to position.

5.Enter the label Position.

6. Click next, next, next and save.

The screenshot shows the Salesforce Object Manager interface. The left sidebar has a 'Fields & Relationships' section with links to Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, and Related Lookup Filters. The main content area is titled 'Fields & Relationships' and shows a table with 7 items, sorted by Field Label. The columns are FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The table includes rows for Created By (CreatedBy), Description (Description__c), Job Posting Site URL (Job_Posting_Site_URL__c), Last Modified By (LastModifiedById), Owner (OwnerId), Site Name (Name), and Status (Status__c). The 'INDEXED' column contains checkmarks for Owner, Site Name, and Status.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Description	Description__c	Checkbox		
Job Posting Site URL	Job_Posting_Site_URL__c	URL(255)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Site Name	Name	Text(80)		✓
Status	Status__c	Picklist		

Create New Field For Review

1. Select Picklist as the Data Type and click Next.

For Field Label enter Core Competencies.

Select Enter values, with each value separated by a new line and enter these values:

- 1
- 2
- 3
- 4
- 5

For Help Text, enter "For this category, rate candidate on a scale of 1 (lowest) to 5 (highest)".

Click Next, Next, then Save & New.

Follow above steps and create two more pick list with Leadership Skills and Experience as the field labels and values same above.

2. Select the Text Area as the Data Type, then click Next.

For Field Label, enter Core Competencies Comments.

Click Next, Next, then Save & New.

Follow above steps and create two more text areas with Leadership Skills Comments and Experience comments as the field labels.

3. Select Checkbox as the Data Type and click Next. For Field Label, enter Recommend for Hire.

For Help Text, enter "Do you recommend that we hire this candidate? "

Click, Next , next then save & new.

4. Select the Text Area as the Data Type and click Next.

For Field Label, enter Reason Recommended.

Click Next, Next, then Save & New.

5. Create a lookup relationship field for Interviewer.

Select Lookup Relationship as the Data Type and click Next.

For Related To, select Interviewer and click Next.

For field label , enter interviewer and click next, next, next then save and new.

6.Create a master-detail relationship field for Job Application.

Select Master-Detail Relationship as the Data Type and click Next.

For Related to, select Job Application and Click Next.

For Field Label, enter Job Application.

Click next, next, next and save.

The screenshot shows two views of the Salesforce Object Manager interface. The top view displays a list of field types: Currency, Date, Date/Time, Email, Geolocation, Number, Percent, Phone, Picklist (selected), Picklist (Multi-Select), Text, Text Area, and Text Area (Long). The bottom view shows the configuration for a 'Job Posting Site' object. The 'Fields & Relationships' tab is selected. In the 'Field Label' section, 'Job Application' is entered. Under 'Values', 'Use global picklist value set' is selected. Other options include 'Enter values, with each value separated by a new line' and a dropdown menu. Below these are checkboxes for 'Display values alphabetically, not in the order entered' and 'Use first value as default value'. A checked checkbox 'Restrict picklist to the values defined in the value set' is also present. The 'Field Name' is left empty, and the 'Description' and 'Help Text' fields are also empty. Navigation buttons 'Previous', 'Next', and 'Cancel' are visible at the top right of the configuration pane. The status bar at the bottom indicates the date and time as 10/31/2023 at 5:41 PM.

The screenshot shows the Salesforce Setup interface. The user is navigating through the Object Manager for the 'Job Posting' object. They have selected the 'Fields & Relationships' tab. A new custom field, 'Job Posting Site', is being created. The field information includes:

- Field Label:** Job Posting Site
- Field Name:** Job_Posting_Site
- API Name:** Job_Posting_Site_c
- Description:** Help Text
- Data Owner:** Field Usage
- Data Sensitivity Level:** Data Sensitivity Level
- Compliance Categorization:** Compliance Categorization
- Created By:** HARUN A.M. 21/10/2023, 10:46 am
- Modified By:** HARUN A.M. 21/10/2023, 10:46 am
- Object Name:** Job Posting
- Data Type:** Master-Detail

Create A Relationships Object

Creating a master-detail relationship between Reviews and job posting site.

1. From setup, click object manager.
2. Select Reviews object, click on field and relationships, click new.
3. Select the data type as Master-detail relationship.
4. Click Next, relate to the Job posting site.
5. Enter the label Job Posting site.
6. Click next, next, next and save.

The screenshot shows the Salesforce Setup interface. The user is navigating through the Object Manager for the 'Review' object. They have selected the 'Fields & Relationships' tab. A new relationship, 'Job Posting Site', is being created. The relationship details are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Job Posting Site	Job_Posting_Site_c	Master-Detail(Job Posting Site)		✓
Last Modified By	LastModifiedBy	Lookup(User)		
Review Number	Name	Auto Number		✓

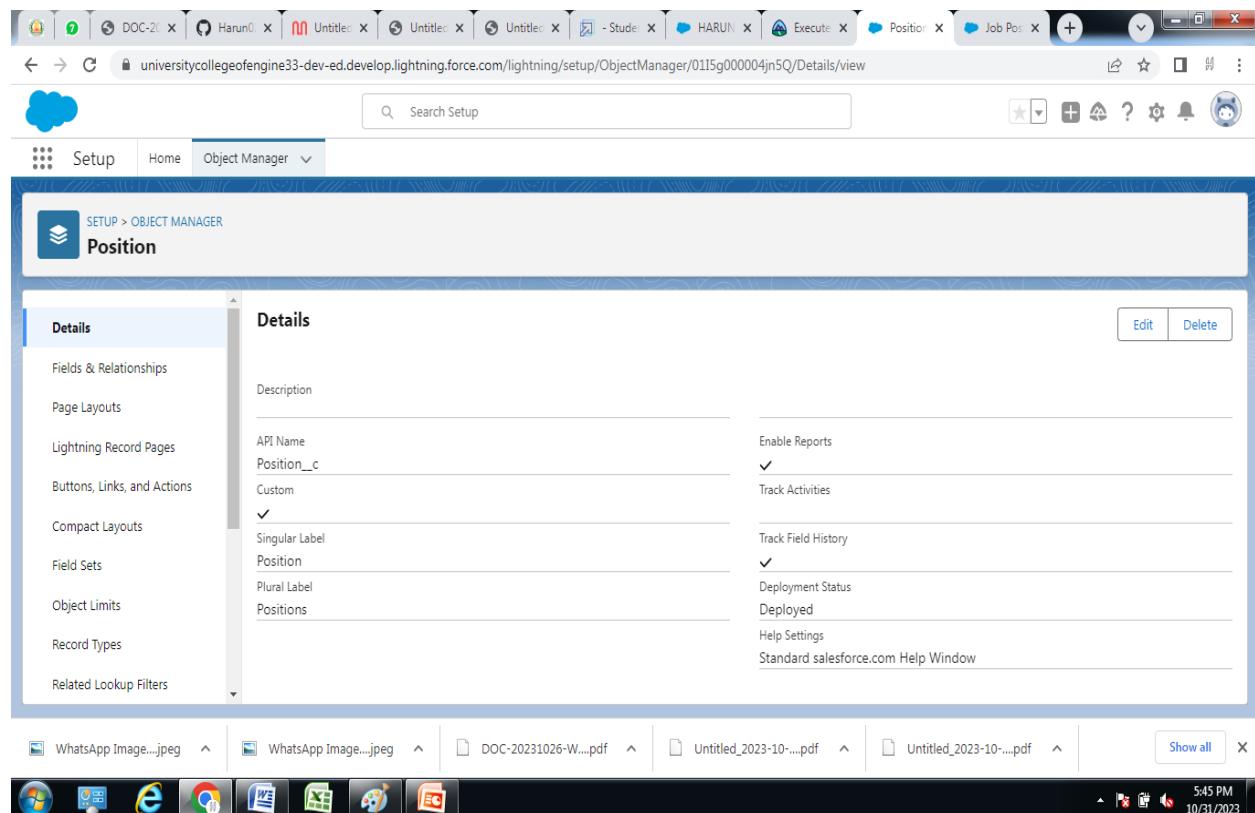
Milestone-7

Page Layout

In Salesforce, a page layout is a visual design of a page that determines the organization and arrangement of fields, buttons, and other components on a page. Page layouts can be customized to show the fields and related information that are most relevant to different users, roles, and record types. They can also be used to control the visibility and access to fields, buttons, and other components on a page.

Modifying The Page Layouts

1. From setup, click on object manager.
2. Click position, then page lay out.
3. Click down array next to the position layout and select layout.
4. Scroll down to the job posting related list, and click the wrench icon in the header to edit it.
5. From the available fields section, select
 - Job posting site:Status Job.
 - posting site: technical site.
- 6.Click add.
7. From the selected field selection , select job posting: Job posting number and click remove.
8. Click Ok, then save.



The screenshot shows the Salesforce Object Manager interface for the 'Position' object. The left sidebar lists various setup categories, with 'Page Layouts' selected. The main content area displays a table titled 'Page Layouts' with two items:

PAGE LAYOUT NAME	CREATED BY	MODIFIED BY
Position	HARUN A M, 21/10/2023, 11:23 am	HARUN A M, 21/10/2023, 11:26 am
Position Layout	HARUN A M, 21/10/2023, 11:22 am	HARUN A M, 21/10/2023, 11:22 am

The browser address bar shows the URL: <https://universitycollegeofengine33-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01I5g000004jn5Q/PageLayouts/view>. The system navigation bar at the bottom includes icons for Home, Setup, and various reports.

This screenshot is identical to the one above, showing the 'Position' page layout in the Salesforce Object Manager. The left sidebar shows 'Page Layouts' is selected. The main content area displays the same table of page layouts:

PAGE LAYOUT NAME	CREATED BY	MODIFIED BY
Position	HARUN A M, 21/10/2023, 11:23 am	HARUN A M, 21/10/2023, 11:26 am
Position Layout	HARUN A M, 21/10/2023, 11:22 am	HARUN A M, 21/10/2023, 11:22 am

The browser address bar shows the URL: <https://universitycollegeofengine33-dev-ed.lightning.force.com/lightning/setup/ObjectManager/01I5g000004jn5Q/PageLayouts/view>. The system navigation bar at the bottom includes icons for Home, Setup, and various reports.

Milestone-8

Validation Rules

A validation rule is a process which checks out (validate) the inputs given by any user is correct or not according to your requirement.

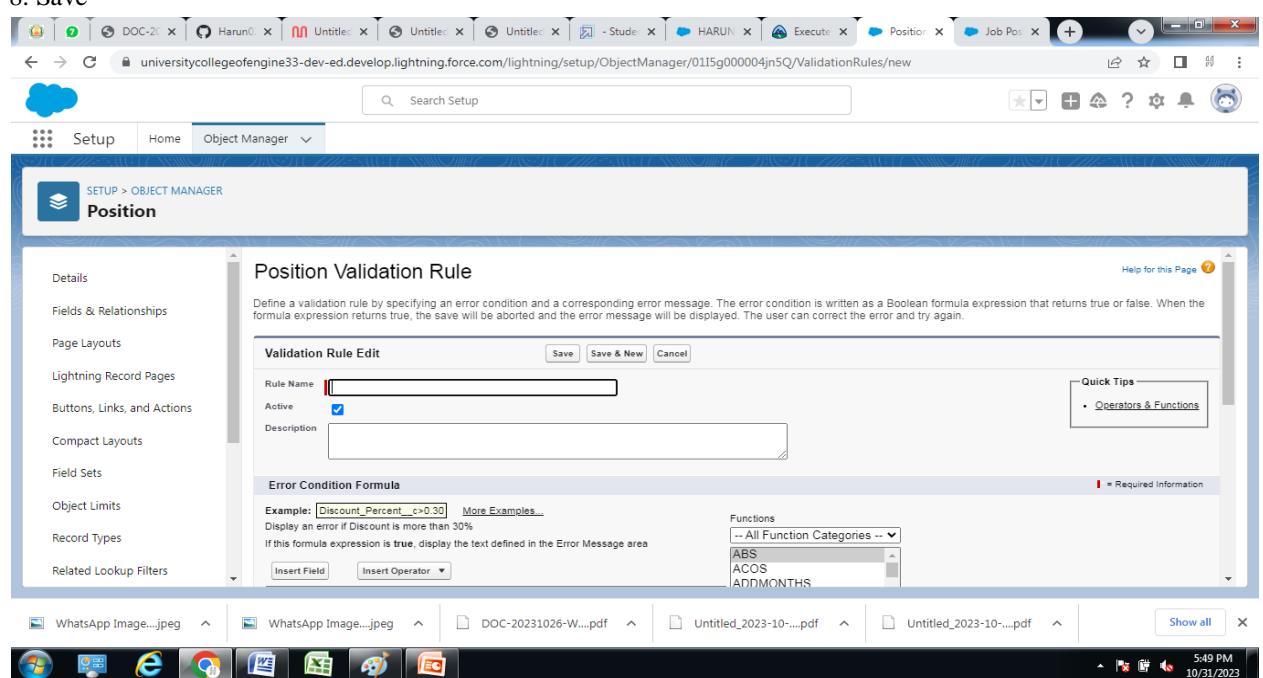
Creating A Validation Rule

To create a validation rule:

Go to object manager, select the object on which validation rule has to be implemented, scroll down and click validation rule, New.

Give details as:

- 1..Rule name: Phone number validation rule.
2. Active: checked.
- 3.Description: phone number should not be more than or less than 10 digits.
- 4.Under Error Condition Formula: write the condition using insert field, insert operator, insert function ----
NOT(OR(REGEX(Phone__c, "^[0-9]{10}")))
- 5.Using check syntax: check if the formula you entered is valid or not.
- 6.Error Message: Please give a valid phone number
- 7.Error location: select field
8. Save

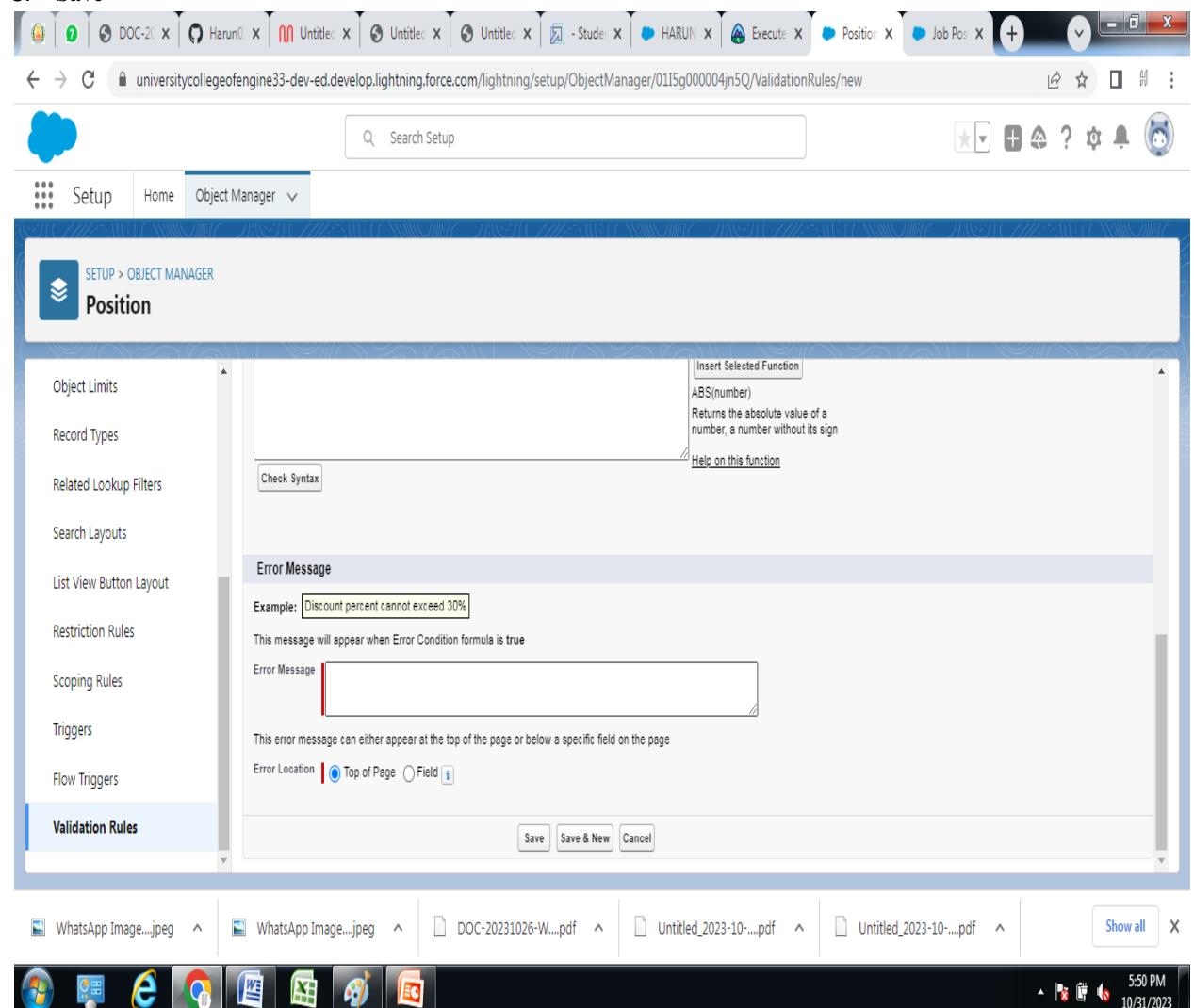


Activity 2

1. Rule name: Phone number validation rule.
2. Active: checked.
3. Description: phone number should not be more than or less than 10 digits.
4. Under Error Condition Formula: write the condition using insert field, insert operator, insert function ----
NOT(OR(REGEX(Phone__c, "^[0-9]{10}"))).
5. Using check syntax: check if the formula you entered is valid or not.
6. Error Message: Please select check box of technical site.

7. Error location: select field (Technical site)

8. Save



Milestone-9

Profile

A profile is a group/collection of settings and permissions that define what a user can do in salesforce. A profile controls “Object permissions, Field permissions, User permissions, Tab settings, App settings, Apex class access, Visualforce page access, Page layouts, Record Types, Login hours & Login IP ranges.

A profile can be assigned to many users, but a user can be assigned to a single profile at a time.

Creation On Profile

From Setup enter Profiles in the Quick Find box, and select Profiles.

1. From the list of profiles, find Standard User.
2. Click Clone.
3. For profile name, enter Event user profile.
4. Click save.
5. While still on the Event profile page, then click edit.
6. Scroll down Custom Object Permissions and Give view all access permissions to the Order details, supplier, product, customer, category, payment.

Screenshot of the Salesforce Setup Home page.

The URL in the browser is <https://universitycollegeofengine33-dev-ed.lightning.force.com/lightning/setup/SetupOneHome/home>.

The page features a search bar at the top with the placeholder "Search Setup". Below it is a navigation bar with "Setup", "Home", and "Object Manager".

A sidebar on the left shows a search field with "profiles" typed in, a "Users" section, and a "Profiles" section which is currently selected. A message says "Didn't find what you're looking for? Try using Global Search.".

The main content area is titled "SETUP Home". It contains three cards:

- Get Started with Einstein Bots**: Launch an AI-powered bot to automate your digital connections. Includes a "Get Started" button.
- Mobile Publisher**: Use the Mobile Publisher to create your own branded mobile app. Includes a "Learn More" button.
- Real-time Collaborative Docs**: Transform productivity with collaborative docs, spreadsheets, and slides inside Salesforce. Includes a "Get Started" button.

Below these cards is a section titled "Most Recently Used" with a list of recently used URLs.

Screenshot of the Salesforce Profiles page.

The URL in the browser is <https://universitycollegeofengine33-dev-ed.lightning.force.com/lightning/setup/EnhancedProfiles/page?address=%2F00e5g000006a9pe%2Fe%3DfretURL%3D%252F00...>.

The page title is "Profiles".

The main content is a "Profile Edit" form for the "Standard User" profile. The "Name" field is "Standard User" and the "User License" is "Salesforce". There is a "Custom Profile" checkbox.

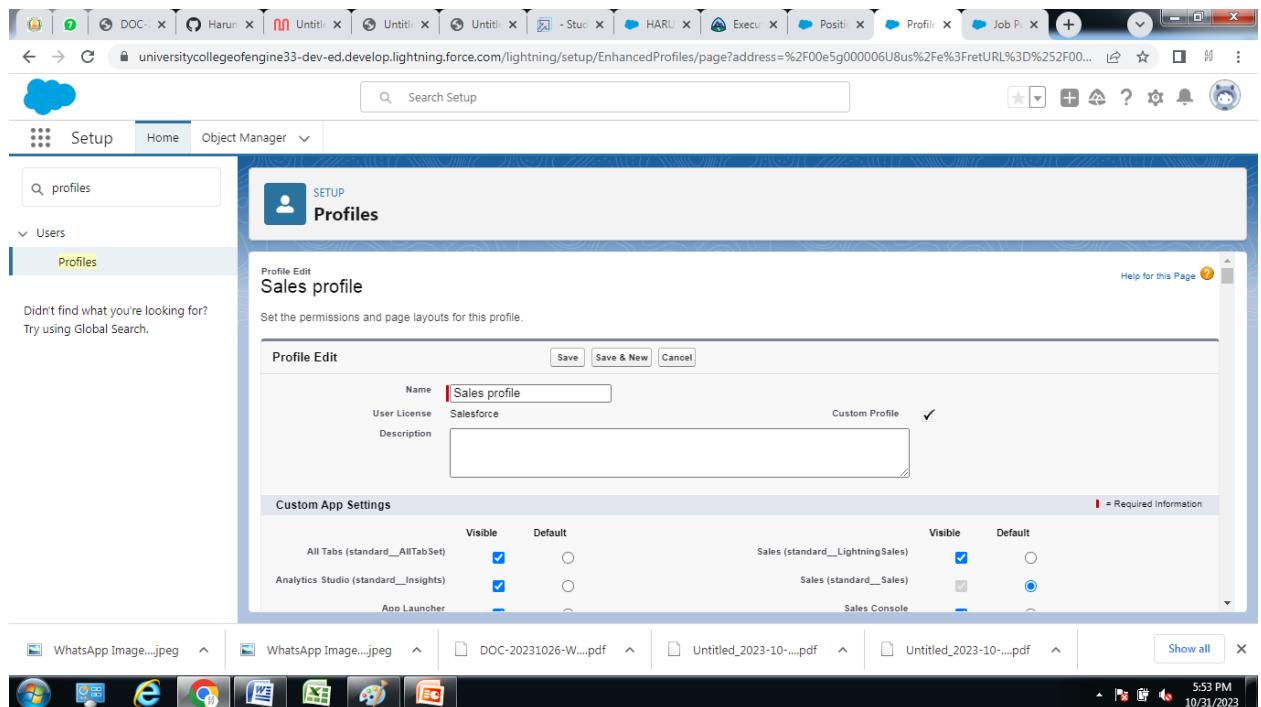
The "Custom App Settings" section lists various apps with checkboxes for "Visible" and "Default".

App	Visible	Default	App	Visible	Default
All Tabs (standard__AllTabSet)	<input checked="" type="checkbox"/>	<input type="radio"/>	Sales (standard__LightningSales)	<input checked="" type="checkbox"/>	<input type="radio"/>
Analytics Studio (standard__Insights)	<input checked="" type="checkbox"/>	<input type="radio"/>	Sales (standard__Sales)	<input checked="" type="checkbox"/>	<input checked="" type="radio"/>
App Launcher (standard__AppLauncher)	<input checked="" type="checkbox"/>	<input type="radio"/>	Sales Console (standard__LightningSalesConsole)	<input checked="" type="checkbox"/>	<input type="radio"/>
Bolt Solutions (standard__LightningBolt)	<input checked="" type="checkbox"/>	<input type="radio"/>	Salesforce Chatter (standard__Chatter)	<input checked="" type="checkbox"/>	<input type="radio"/>
Community (standard__Community)	<input checked="" type="checkbox"/>	<input type="radio"/>	Salesforce Scheduler Setup (standard__LightningScheduler)	<input type="radio"/>	<input type="radio"/>
Content (standard__Content)	<input checked="" type="checkbox"/>	<input type="radio"/>	Sample Console (standard__ServiceConsole)	<input type="radio"/>	<input type="radio"/>
Data Manager (standard__DataManager)	<input checked="" type="checkbox"/>	<input type="radio"/>	Service (standard__Service)	<input checked="" type="checkbox"/>	<input type="radio"/>
Digital Experiences (standard__SalesforceCMS)	<input checked="" type="checkbox"/>	<input type="radio"/>	Service Console (standard__LightningService)	<input checked="" type="checkbox"/>	<input type="radio"/>

Below the form is a "Most Recently Used" section with a list of URLs.

Activity 2

Create a profile with the profile name as “Sales profile”



Milestone-10

User

A user is anyone who logs in to Salesforce. Users are employees at your company, such as sales reps, managers, and IT specialists, who need access to the company's records. Every user in Salesforce has a user account. The user account identifies the user, and the user account settings determine what features and records the user can access.

Creating A User

From setup type “users” in quick find and select users, then click New User

- First Name: Sanjay.
- Last Name: Gupta.
- Alias: Sanj.
- Email: provide your personal email id for future reference.
- Username: sanjaygupta@thesmartbridge.com.
- Nickname: Sanju.
- Role: leave it as default.
- User license salesforce:.
- Profile: Event user profile.

Screenshot of the Salesforce Setup interface showing the 'Users' page.

The left sidebar shows the navigation menu under 'Users':

- Permission Set Groups
- Permission Sets
- Profiles
- Public Groups
- Queues
- Roles
- User Management Settings
- Users** (selected)
- Feature Settings
- Data.com
- Prospector Users

The main content area displays the 'All Users' table:

Action	Full Name	Alias	Username	Role	Active	Profile
Edit	A.M_HARUN	HARUN	harunam@company.sandbox		<input checked="" type="checkbox"/>	System Administrator
Edit	Brown_Courtney	cbow	cbrown@apblue0562419.com	SVP_Sales & Marketing	<input checked="" type="checkbox"/>	Standard Platform User
Edit	Chatter_Expert	Chatter	chatty00d5g00000kgyudeaz0p9xkjg8ch@chatter.salesforce.com		<input checked="" type="checkbox"/>	Chatter Free User
Edit	Garapati_Abhilash	abhi	abhilash60@gmail.com		<input checked="" type="checkbox"/>	Cross Org Data Proxy User
Edit	Gupta_Sanjay	Sanj	sanjay37gupta@thesmartbridge.com		<input checked="" type="checkbox"/>	Cross Org Data Proxy User
Edit	User_Integration	integ	integration00d5g00000kgyudeaz.com		<input checked="" type="checkbox"/>	Analytics Cloud Integration User
Edit	Yashika_Rajput	yashika	yashika.rajput@thesmartbridge.com		<input checked="" type="checkbox"/>	Analytics Cloud Security User

At the bottom of the page, there is a link to 'Help for this Page'.

Screenshot of the Salesforce Setup interface showing the 'User Edit' page for 'Sanjay Gupta'.

The left sidebar shows the navigation menu under 'Users':

- Permission Set Groups
- Permission Sets
- Profiles
- Public Groups
- Queues
- Roles
- User Management Settings
- Users** (selected)
- Feature Settings
- Data.com
- Prospector Users

The main content area displays the 'User Edit' form for 'Sanjay Gupta':

User Edit
Sanjay Gupta

User Edit (Save, Save & New, Cancel)

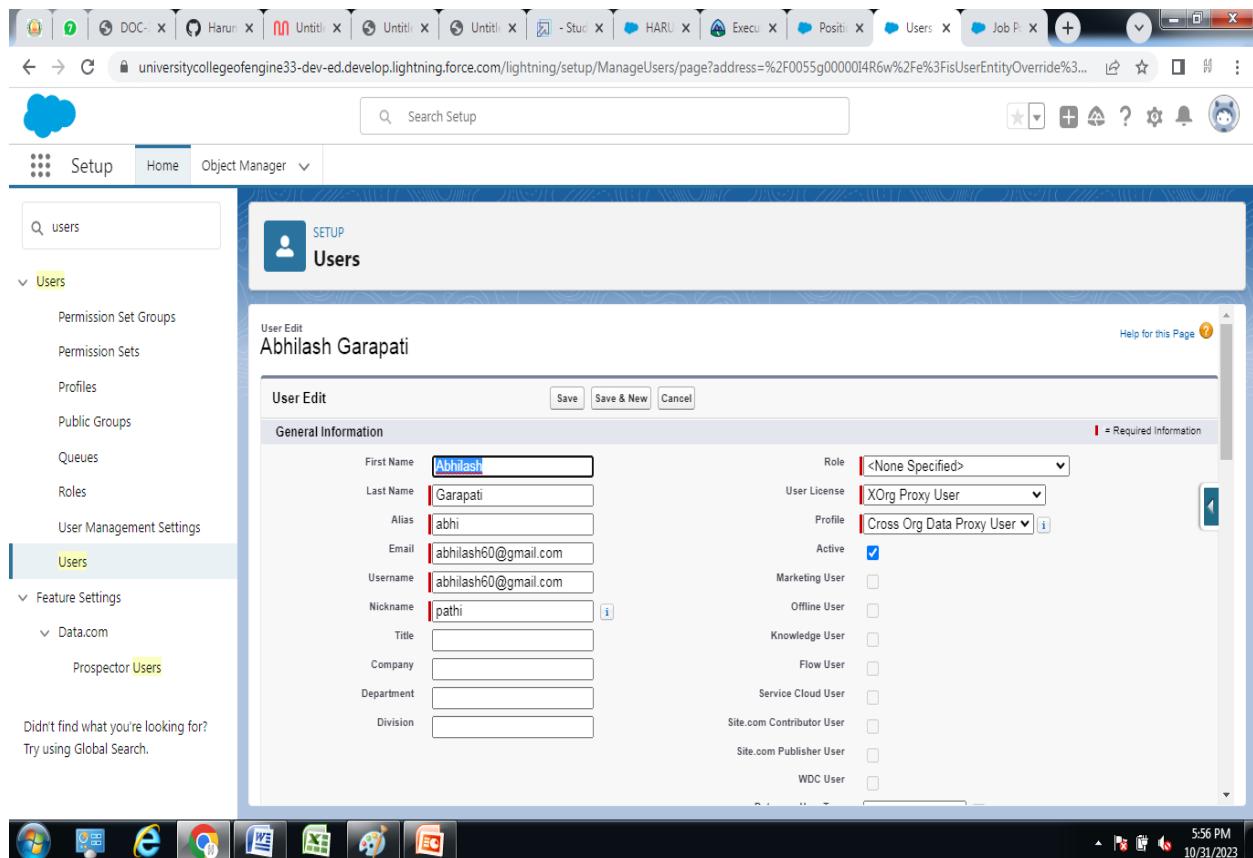
General Information (Required Information)

First Name	Sanjay	Role	<None Specified>
Last Name	Gupta	User License	XOrg Proxy User
Alias	Sanj	Profile	Cross Org Data Proxy User
Email	sanjay37@gmail.com	Active	<input checked="" type="checkbox"/>
Username	sanjay37gupta@thesmartbri	Marketing User	<input type="checkbox"/>
Nickname	Sanju	Offline User	<input type="checkbox"/>
Title		Knowledge User	<input type="checkbox"/>
Company		Flow User	<input type="checkbox"/>
Department		Service Cloud User	<input type="checkbox"/>
Division		Site.com Contributor User	<input type="checkbox"/>

At the bottom of the page, there is a link to 'Help for this Page'.

Activity 2

Create a user with a username as “Abhilash Garapati”, and assign him the sales profile.



Milestone-11

Permission Set

In Salesforce, a permission set is a collection of settings and permissions that give users access to various tools and functionality in the platform. Permission sets can be used to grant additional access to users beyond what is included in their profile, without modifying the profile itself. This allows for granular control over user access and permissions within the Salesforce environment. Permission sets can be assigned to individual users or to a group of Users.

Creating A Permission Set:

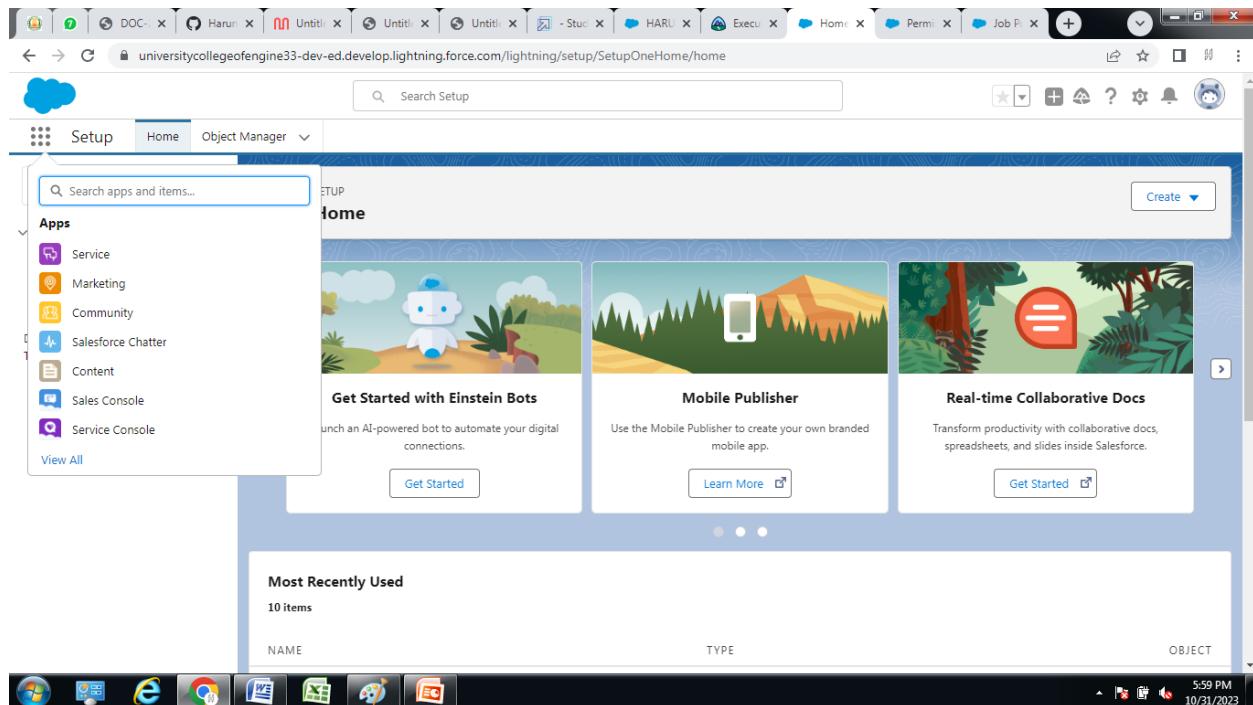
1. From setup search “permission sets” in quick find and select permission set then click on New.
2. Enter label as: HR Recruiter Supplier Permits and Save.
3. After saving the permission click on the Manage assignment.
4. Click on edit and give Object permission (Edit) then save.
5. Click on edit, and give object permission (Edit) then save.
6. Now click on the Add Assignment.
7. Now select the users and click on save.

The screenshot shows the Salesforce Setup interface under the 'Permission Sets' tab. The page title is 'Permission Sets'. A search bar at the top right contains the text 'Search Setup'. Below the search bar, there are tabs for 'Setup', 'Home', and 'Object Manager'. A sidebar on the left has a 'Users' section with a 'Permission Sets' link, which is highlighted in yellow. The main content area displays a table titled 'Permission Sets' with columns: Action, Permission Set Label, Description, and License. The table lists various permission sets such as 'Buyer', 'Buyer Manager', 'C360 High Scale Flow Integration User', 'CRM User', etc. At the bottom of the table, there are pagination controls showing '1-30 of 30' and '0 Selected'. A help link 'Help for this Page' is located in the top right corner.

The screenshot shows the Salesforce Setup interface under the 'Permission Sets' tab. The page title is 'Permission Sets'. A search bar at the top right contains the text 'Search Setup'. Below the search bar, there are tabs for 'Setup', 'Home', and 'Object Manager'. A sidebar on the left has a 'Users' section with a 'Permission Sets' link, which is highlighted in yellow. The main content area displays a dialog box titled 'Clone: HR Recruiter Supplier Permits'. The dialog is for creating a new permission set based on an existing one. It has fields for 'Label' (set to 'HR_Recruiter_Supplier_Permits'), 'API Name' (set to 'HR_Recruiter_Supplier_Permits'), 'Description' (empty), and 'Session Activation Required' (unchecked). There is also a 'License' section which is currently empty. At the bottom of the dialog are 'Save' and 'Cancel' buttons. A help link 'Help for this Page' is located in the top right corner.

Create A Record (Positions)

- Click on app launcher on left side of the screen.
- Search recruiting and click on it.
- Click on positions tabs.
- Click new and fill details and save.



Milestone-12

Reports

A report is a list of records that meet the criteria you define. It's displayed in rows and columns, and can be filtered, grouped, or displayed in a graphical chart. Every report is stored in a folder. Folders can be public, hidden, or shared and can be set to read-only or read/write.

There are 4 types of report formats in Salesforce:

1. Tabular Reports:

This is the most basic report format. It just displays the row of records in a table with a grand total. While easy to set up they can't be used to create groups of data or charts and also cannot be used in Dashboards. They are mainly used to generate a simple list or a list with a grand total.

2. Summary Reports:

It is the most commonly used type of report. It allows grouping of rows of data, view subtotal, and create charts.

3. Matrix Report:

It is the most complex report format. Matrix report summarizes information in a grid format. It allows records to be grouped by both columns and rows. It can also be used to generate dashboards. Charts can be added to this type of report.

4. Joined Reports:

These types of reports let us create different views of data from multiple report types. The data is joined reports are organized in blocks. Each block acts as a subreport with its own fields, columns, sorting, and filtering. They are used to group and show data from multiple report types in different views. **Report types:** Report type determines which set of records will be available in a report. Every report is based on a particular report type. The report type is selected first when we create a report. Every report type has a primary object and one or more related objects. All these objects must be linked together either directly or indirectly.

A report type cannot include more than 4 objects.

Once a report is created its report type cannot be changed.

There are 2 types of report types:

a. Standard Report Types:

Standard Report Types are automatically included with standard objects and also with custom Objects where "Allow Reports" is checked. Standard report types cannot be customized and automatically include

standard and custom fields for each object within the report type. Standard report types get created when an object is created, also when a relationship is created.

b. Custom Report Types:

Custom report types are reporting templates created to streamline the reporting process. Custom Reports are created by an administrator or User with “Manage Custom Report Types” permission. Custom report types are created when standard report types cannot specify which records will be available on reports. In custom report types we can specify objects which will be available in a particular report. The primary object must have a relationship with other objects present in a report type either directly or indirectly.

Creating A Report

1. From the reports tab, click new report.
2. Select the report type Job application with position for the report, and click Create.
3. Customize your report accordingly and include all fields, Reports need to be Grouped by one field.(ex - Created by)(require to enable add chart) Then save (Job application with position) or run it.

The screenshot shows the Salesforce Lightning interface with the following details:

- Header:** Shows a toolbar with various icons, a browser address bar with the URL "universitycollegeofengine33-dev-ed.lightning.force.com/lightning/o/Report/home?queryScope=mru", and a top navigation bar with Sales, Home, Opportunities, Leads, Tasks, Files, Accounts, Contacts, Campaigns, Dashboards, Reports (selected), Chatter, Groups, and More.
- Left Sidebar:** A sidebar menu with sections: Recent, REPORTS, FOLDERS, and FAVORITES. Under REPORTS, there are categories: Recent, Created by Me, Private Reports, Public Reports, All Reports, and FLOWs. Under FOLDERS, there are categories: All Folders, Created by Me, Shared with Me. Under FAVORITES, there is a category: All Favorites.
- Content Area:** The main area displays a table of recent reports. The columns are: Report Name, Description, Folder, Created By, Created On, and Subscribed. The table includes the following rows:

REPORTS	Report Name	Description	Folder	Created By	Created On	Subscribed
Recent	Case Status	Private Reports	HARUN A M	23/10/2023, 7:47 pm		
Created by Me	Opportunities to Work	Private Reports	HARUN A M	23/10/2023, 7:42 pm		
Private Reports	Job posting sites	Private Reports	HARUN A M	21/10/2023, 1:21 pm		
Public Reports	Lead Source Report	Private Reports	HARUN A M	23/10/2023, 7:30 pm		
All Reports	Lead Source Report	Private Reports	HARUN A M	23/10/2023, 7:27 pm		
FLOWs	Sample Flow Report: Screen Flows	Public Reports	Automated Process	25/9/2023, 6:14 pm		
All Folders	Lead Source Report	Private Reports	HARUN A M	22/10/2023, 9:43 pm		
Created by Me	Web Leads	Private Reports	HARUN A M	22/10/2023, 9:40 pm		
Shared with Me	Job application with position	Private Reports	HARUN A M	21/10/2023, 1:19 pm		
FAVORITES	New Opportunities with Partners Report	Private Reports	HARUN A M	21/10/2023, 1:08 pm		
All Favorites						
- Bottom:** A taskbar with icons for File, Home, Recent, and Help, followed by a URL bar with "https://universitycollegeofengine33-dev-ed.lightning.force.com/lightning..." and a system status bar showing "6:01 PM 10/31/2023".

The screenshot shows the Salesforce Lightning interface. At the top, there's a toolbar with various icons and a search bar. Below the toolbar is a navigation bar with links like Sales, Home, Opportunities, Leads, Tasks, Files, Accounts, Contacts, Campaigns, Dashboards, Reports, Chatter, Groups, and More. A blue bar indicates the current section is Reports.

The main area is titled "REPORT" and shows a preview for "Job application with position". There are two tabs: "Campaigns with Opportunities" (selected) and "Job application with position".

The report preview area has a sidebar on the left labeled "Fields" with sections for "Groups" and "Columns". Under "Groups", there are filters for Start Date, Campaign ID, and Campaign Name. Under "Columns", there are filters for Opportunity Name, fx Height, and Campaign Description. The main preview area says "No records returned. Try editing report filters:" and lists two items:

- Set the Start Date filter to All Time.
- Edit other filters in the filter panel.

At the bottom of the preview area, there are buttons for Row Counts, Detail Rows, Subtotals, Grand Total, and Conditional Formatting. The status bar at the bottom right shows the time as 6:01 PM and the date as 10/31/2023.

Milestone-13

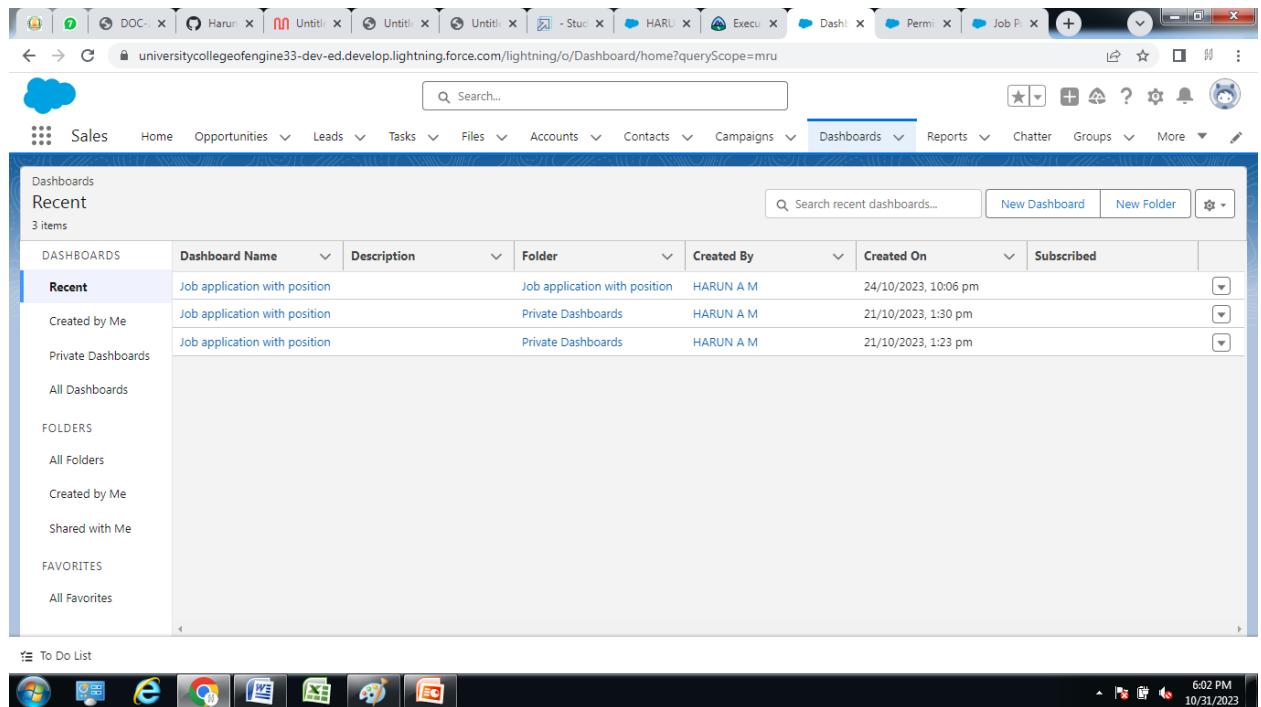
Dashboard

Dashboards provide more insights than reports as they combine the data from many reports and show a summarized result. Looking at many reports at a time gives the flexibility of combining the results from them quickly. Also summaries in dashboards help us decide on action plans quicker.

The dashboards can contain charts, graphs and Tabular data.

Create A Dashboard

- Click the Dashboards tab.
- Click New Dashboard.
- Name the dashboard Job application with position and click Create.
- Click +Component.
- Select Job application with position and click Select.
- Select the Vertical Bar Chart component and click Add.
- Click Save and then Done.



Milestone-14

Report

- Click on App Launcher on left side of screen.
- Search Recruiting & click on it.
- Click on Reports Tab.
- Click on Job application with position & see records.

Dashboard

- Click on App Launcher on left side of screen.
- Search Recruiting & click on it.
- Click on Dashboard Tab.
- Click on Job application with position & see records.

Milestone-15

Apex Triggers

Apex is a coding language of Salesforce. It can be invoked or started using triggers. A trigger is a set of Apex code that runs before or after data manipulation language (DML) events.

A DML event could be a variety of data processing tasks that include the standard insert, update, and delete commands. With Apex triggers, you can automate tasks that would otherwise be nearly impossible to accomplish using only the Salesforce user interface. Triggers enable you to create custom scripts that you can implement according to your needs, and the only limitation is your coding skills.

There are two Salesforce Apex trigger types:

1.Before triggers. These are helpful in cases that require a validation process before accepting a change. They run before any database changes.

2.After triggers. These are helpful in cases where you need to modify your database records and when the necessary value is stored in other records. They run after any database changes. Both types will help you perform custom tasks and manage records effectively. They can help you perform bulk actions, as they can handle several records simultaneously.

Creation Of The Trigger

Use Case:

HR is struggling!! Not knowing whom to hire on priority so she contact developer. Now the developer made the HR task bit easy with this changes . There is the Review__c Object and there is 2 field priority and Recommended for Hire(CheckBox) so condition is like if suppose if we checked the checkbox than priority should be high **Code lines**

```
trigger PriorityTrig on Review__c (before insert) {  
List<Review__c> myList = trigger.new;  
for(Review__c rv:myList){  
if(rv.Recommendednd_For_Hire__c == true){  
rv.Priority__c= 'high'; system.debug(rv);  
}  
}  
}  
}
```

The screenshot shows the Salesforce Developer Console interface. At the top, it says "Developer Console - Google Chrome" and the URL "d2w00000oeey2eaf-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage". Below that is a menu bar with File, Edit, Debug, Test, Workspace, Help, and a Go To button. The main area contains the trigger code, which is identical to the one provided above. The code editor has line numbers from 1 to 16. Below the code editor is a navigation bar with tabs: Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The Problems tab is currently selected. A search bar and a toolbar with various icons are at the bottom.

```
1 trigger PriorityTrig on Review__c (before insert) {  
2  
3     List<Review__c> myList = trigger.new;  
4  
5     for(Review__c rv:myList){  
6         if( rv.Recommendednd_For_Hire__c == true){  
7             rv.Priority__c= 'high';  
8             system.debug(rv);  
9         }  
10    }  
11 }  
12  
13  
14  
15  
16
```

File	Line	Iteration
ApexTriggers	1	1
VerifyDate	1	1
FlightSystemsChecklist	1	1
EmailMissionSpecialist	12	1

Milestone-16

DML Operations And Batchable

Create and modify records in salesforce in using the data manipulation language, abbreviated as DML. DML provides a straightforward way to manage records by providing simple statements to insert, update, merge, delete, and restore records.

Because Apex is a data-focused language and is saved on the Lightning Platform, it has direct access to your data in Salesforce. Unlike other programming languages that require additional setup to connect to data sources, with Apex DML, managing records is made easy! By calling DML statements, you can quickly perform operations on your Salesforce records. This example adds the Acme account to Salesforce. An account sObject is created first and then passed as an argument to the insert statement, which persists the record in Salesforce.

Apex Batch:

Apex Batch For Fetching the records of job posting site. Batch class in salesforce is used to run large jobs (think thousands or millions of records!) that would exceed normal processing limits. Using Batch Apex, you can process records asynchronously in batches. If you have a lot of records to process, for example, data cleansing or archiving. A Batch class allows you to define a single job that can be broken up into manageable chunks that will be processed separately.

Create The DML Insert For The Job Posting Site

The HR Manager was looking in the Application and just checking how many candidates have applied for the job and he came to know that there are too many Candidates who applied for the job. So what the HR manager is Doing whenever there are too many Records he fetches the information. So he went to the Developer and asked to fetch the Record of the Job Posting Site. So This Task will be Executed with Apex Batch.

Benefits:

If suppose Records are getting failed after fetching the 100 Records, so we must try again to execute the Batch Method so it wont start with the starting 100 records it start fetching the records from 101 as batch runs Asynchronously.

The screenshot shows the Salesforce Developer Console interface. At the top, there's a navigation bar with links like File, Edit, Debug, Test, Workspace, Help, and a Go To button. Below the navigation bar, the main area displays a code editor with Java-like syntax highlighting. The code is a class named 'InsertMultipleRec' with a static method 'InsertMethod'. This method creates a list of 'Job_Posting_Site__c' records, each with 'Name' set to 'naukri' and 'status__c' set to 'Complete'. It then adds these records to the list and performs an insert operation. A 'system.debug' statement is included to log the inserted records. Below the code editor is a log viewer. The 'Logs' tab is selected, showing a table of log entries. The table has columns for User, Application, Operation, Time, Status, Read, and Size. There are six entries listed, all from the user 'shivam upadhye'. The first entry is a success (Status: Success, Read: Read, Size: 722.46 KB). The second entry is also a success (Status: Success, Read: Unread, Size: 589.82 KB). The third entry is a success (Status: Success, Read: Unread, Size: 3.04 KB). The fourth entry is an invalid id (Status: Invalid id: naukri, Read: Unread, Size: 3.72 KB). The fifth entry is an invalid id (Status: Invalid id: naukri.com, Read: Unread, Size: 3.72 KB). The sixth entry is an invalid id (Status: Invalid id: naukri.com, Read: Unread, Size: 3.68 KB). At the bottom of the log viewer is a search bar and a toolbar with various icons. The status bar at the bottom right shows the date and time: 10-05-2023 08:48.

User	Application	Operation	Time	Status	Read	Size
shivam upadhye	Unknown	/services/data/v57.0/tooling/executeA...	10/05/2023, 08:43:18	Success	Read	722.46 KB
shivam upadhye	Unknown	/services/data/v57.0/tooling/executeA...	10/05/2023, 08:40:19	Success	Unread	589.82 KB
shivam upadhye	Browser	/aura	10/05/2023, 08:37:15	Success	Unread	3.04 KB
shivam upadhye	Unknown	/services/data/v57.0/tooling/executeA...	10/05/2023, 08:36:09	Invalid id: naukri	Unread	3.72 KB
shivam upadhye	Unknown	/services/data/v57.0/tooling/executeA...	10/05/2023, 08:35:42	Invalid id: naukri.com	Unread	3.72 KB
shivam upadhye	Unknown	/services/data/v57.0/tooling/executeA...	10/05/2023, 08:30:25	Invalid id: naukri.com	Unread	3.68 KB

```
public class InsertMultipleRec {  
    public static void InsertMethod(){  
        List<Job_Posting_Site__c> insRec = new List <Job_Posting_Site__c>();  
        for(integer i =1; i<1000; i++){  
  
            Job_Posting_Site__c r = new Job_Posting_Site__c();  
  
            r.Name = 'naukri';  
            r.status__c='Complete';  
            insRec.add(r);  
        }  
        insert insRec;  
        system.debug(insRe  
c);  
    }  
}
```

```

1 public class InsertMultipleRec {
2
3 }

```

File	Line	Iteration
ApexTriggers	1	1
VerifyDate	1	1
FlightSystemsChecklist	1	1
EmailMissionSpecialist	12	1

Create The Batch Apex

Create the Batch Apex

```

1 public class Util1 implements Database.Batchable<sObject> {
2
3
4     public Database.QueryLocator start(Database.BatchableContext bc) {
5         string myList = 'SELECT Id, Name, Technical_Site__c FROM Job_Posting_Site__c';
6
7         return Database.getQueryLocator(myList);
8         system.debug('start method');
9     }
10
11    public void execute(Database.BatchableContext bc, List<Job_Posting_Site__c> accList) {
12        system.debug(accList.size());
13        system.debug(accList);
14    }
15
16    public void finish(Database.BatchableContext bc) {
17        system.debug('finish method');
18    }
19
20}
21
22}

```

User	Application	Operation	Time	Status	Read	Size
shivam upadhye	Unknown	Batch Apex	10/05/2023, 09:20:30	Success		3.38 kB
shivam upadhye	Unknown	SerialBatchApexRangeChunkHandler	10/05/2023, 09:20:30	Success		2.46 kB
shivam upadhye	Unknown	SerialBatchApexRangeChunkHandler	10/05/2023, 09:20:30	Success		2.97 kB
shivam upadhye	Unknown	SerialBatchApexRangeChunkHandler	10/05/2023, 09:20:29	Success		2.98 kB
shivam upadhye	Unknown	SerialBatchApexRangeChunkHandler	10/05/2023, 09:20:29	Success	Unread	2.98 kB

```

public class Util1 implements Database.Batchable<sObject> {

    public Database.QueryLocator start(Database.BatchableContext bc) {
        string myList = 'SELECT Id, Name, Technical_Site__c FROM Job_Posting_Site__c';

        return Database.getQueryLocator(myList); system.debug('start
method');
    }

    public void execute(Database.BatchableContext bc, List<Job_Posting_Site__c> accList) {
        system.debug(accList.size()); system.debug(accList);

    }

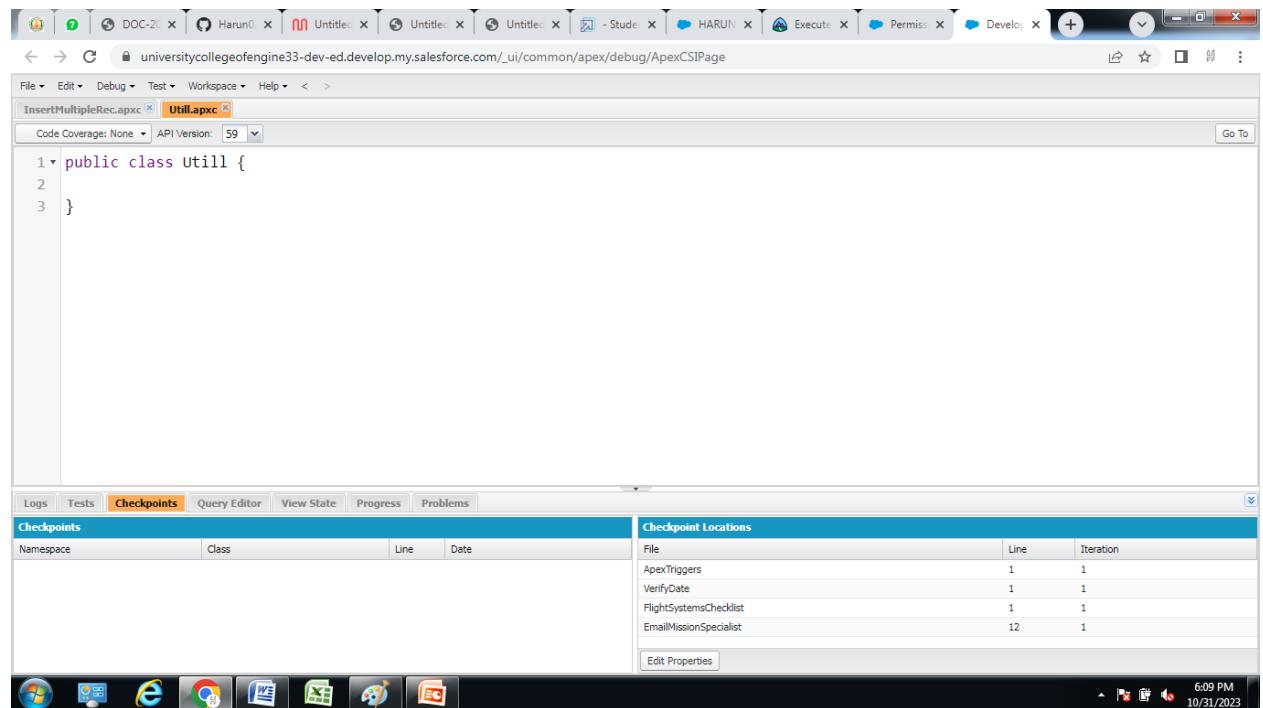
    public void finish(Database.BatchableContext bc) {
        system.debug('finish method');
    }

}

```

Anonymous Code to call the batch class

```
Util1 obj = new Util1(); Database.executeBatch(obj);
```



Screenshot of the Salesforce IDE showing the Apex code editor and checkpoints.

```

1 public class Util1 implements Database.Batchable<sObject> {
2
3     public Database.QueryLocator start(Database.BatchableContext bc) {
4
5         string myList = 'SELECT Id, Name, Technical_Site__c FROM Job_Posting_Site__c';
6
7         return Database.getQueryLocator(myList); system.debug('start method');
8     }
9
10    public void execute(Database.BatchableContext bc, List<Job_Posting_Site__c> accList) { system.debug(accList.size()); system.debug(
11
12    }
13
14
15    public void finish(Database.BatchableContext bc) {

```

The checkpoints tab shows the following data:

Checkpoints			
Namespace	Class	Line	Date
none	ApexTriggers	1	10/31 18:12:01
none	ApexTriggers	1	10/31 18:12:02
none	TestVerifyDate	1	10/31 18:12:00
none	ApexTriggers	1	10/31 18:12:00

Checkpoint Locations		
File	Line	Iteration
ApexTriggers	1	1
VerifyDate	1	1
FlightSystemsChecklist	1	1
EmailMissionSpecialist	12	1

Screenshot of the Salesforce IDE showing the Apex code editor and checkpoints, with an 'Enter Apex Code' dialog open.

```

1 public class Util1 implements Database.Batchable<sObject> {
2
3     public Database.QueryLocator start(Database.BatchableContext bc) {
4
5         string myList = 'SELECT Id, Name, Technical_Site__c FROM Job_Posting_Site__c';
6
7         return Database.getQueryLocator(myList); system.debug('start method');
8     }
9
10    public void execute(Database.BatchableContext bc, List<Job_Posting_Site__c> accList) { system.debug(accList.size()); system.debug(
11
12    }
13
14
15    public void finish(Database.BatchableContext bc) {

```

The checkpoints tab shows the following data:

Checkpoints			
Namespace	Class	Line	Date
none	ApexTriggers	1	10/31 18:12:01
none	ApexTriggers	1	10/31 18:12:02
none	TestVerifyDate	1	10/31 18:12:00
none	ApexTriggers	1	10/31 18:12:00

Checkpoint Locations		
File	Line	Iteration
ApexTriggers	1	1
VerifyDate	1	1
FlightSystemsChecklist	1	1
EmailMissionSpecialist	12	1

The 'Enter Apex Code' dialog contains the following code:

```

1 Util1 obj = new Util1(); Database.executeBatch(obj);

```

DOC-20 X Harun0 X Untitled X Untitled X Untitled X - Stude X HARUN X Execute X Permiss X Develop X

← → C universitycollegeofengineering33-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >

Util.apxc *

Code Coverage: None ▾ API Version: 59 ▾ Go To

```

1 public class Util1 implements Database.Batchable<sObject> {
2
3     public Database.QueryLocator start(Database.BatchableContext bc) {
4
5         string myList = 'SELECT Id, Name, Technical_Site_c FROM Job_Posting_Site_c';
6
7         return Database.getQueryLocator(myList); system.debug('start method');
8     }
9
10
11    public void execute(Database.BatchableContext bc, List<Job_Posting_Site_c> accList) { system.debug(accList.size()); system.debug
12
13    }
14
15    public void finish(Database.BatchableContext bc) {

```

Logs Tests Checkpoints Query Editor View State Progress Problems 1

Status	Test Run	Enqueued Time	Duration	Failures	Total	Overall Code Coverage
X	7075g00008DQHU7	Tue Oct 31 2023 18:12:00 GMT...		0	1	Class
X	7075g00008DQJMG	Tue Oct 31 2023 18:13:08 GMT...		0	1	Overall 48%
						AccountAddressTrigger 66% 2/3
						AccountManager 100% 6/6
						AccountProcessor 100% 7/7
						AddPrimaryContact 100% 13/13
						AnimalLocator 100% 11/11

6:13 PM 10/31/2023