LIGHTNING EXPERIENCE

Salesforce introduced the Lightning Experience in 2015, to make the salesforce applications can be accessible from mobile app.

Lightning Experience provides more faster performance than Classic Experience.

It provides attractive look and feel for the user interface, which has been built by using "Component Based Framework", based on the latest Web Standards.

Use Case: Configure a List View Option/Filter for Lead Object, to represent only "Energy Industry Lead Records".

Click on the "Leads" Tab.

1. Click on "List View Controls" button and expand it.
2. Click on "New" button.
3. Enter the "List View Label, and Name".
4. Select the Visibility option.
    1. Visible to All Users. (Select this Option)
    2. Visible Only to Me.
    3. Visible to a Group of Users.
5. Click on "Save" button.
6. Add the Required Filters.
    1. Click on the "Filter Icon".
    2. Click on "Add Filter" link.
    3. Add the Required Filters based on the need.
    Ex: Lead: Industry == 'Energy'
    4. Click on "Add Filter Logic" link, to Change the Filter Logic.
    5. Click on "Save" button.
7. Select the Required Fields to be visible in the Table.
    1. Click on "List View Controls" and expand it.
    2. Click on "Select Fields to Display" menu item.
    3. Select the "Fields to be visible" by using "Add / Remove" buttons.
    4. Arrange the fields in the required order by using "Up/Down" navigation buttons.
    5. Click on "Save" button.

Observation: It will Add a New Filter Option inside the Picklist.

App Manager: App Manager is a Lightning Experience feature, which enables us to manage the Lightning Applications as below.

1. App Manager allows us to view all the applications (Classic + Lightning) available inside the organization at one place.

2. We can create custom lightning applications.

3. We can customize the Standard Lightning Applications based on the need (i.e., we can change the logo of the applications, we can customize the appearance, personalization and branding. We can customize the navigation items, utility items...etc.)

4. We can upgrade the Custom Classic Applications into Lightning Experience (LEX).

Click on "Setup" menu.

1. Search for the option "App Manager" in Quick Find box.
2. View the Applications.

Creating Lightning Application

Click on "Setup" menu.

1. Search for the option "App Manager" in Quick Find Box.
2. Click on "New Lightning App" button.
3. Enter the Application Label, Name and Description.
4. Select the Image to be used as the Logo for the Application, by using "Upload Image" button from the Local Device.
5. Select the Color, by using "Color Picker", to be applied for the Application Tabs.
6. Select the Checkbox "Apply the Custom Color and Image for the Application, instead of the Organization's theme".
7. Click on "Next" button.
8. Select the Personalization Options and Branding, and Form Factors.
9. Click on "Next" button.
10. Select the Navigation Items to be visible on the Navigation Bar, by using "Add / Remove" buttons.
11. Arrange the Tabs in the required order by using "Up / Down" navigation buttons.
12. Click on "Next" button.
13. Select the Utility Items, to be visible on the Application. (Ex: Chatter Feed, News, Flows...etc.)
14. Click on "Next" button.
15. Select the "Profiles", to whom the Application can be accessible by using "Add / Remove" buttons.
16. Click on "Save" button.

Use Case: Create a Lightning Application "Merck Pharma", by using the "Merck Image" as the Logo for the Application. Add the required Navigation Items on the Navigation Bar.

Use Case: Customize the Standard Sales Application, by adding the Custom Image as the Logo, and Change the Name as "Tesla Sales", and Customize the Navigation Items.

Objects: Object represents a table, which is used to store Application / Business / Organization specific data in the form of "Structured Format".

Salesforce provides 2 types of objects.

1. **Standard Objects:** These are the readymade Objects provided by Salesforce by default as part of Salesforce CRM Application.

Ex: Campaign, Lead, Account, Contact, Opportunity, Case, Solution, Order, Contract, Feedback etc.

Note: We can Customize the Standard objects based on the need but we can not remove them from the Organization.

2. **Custom Objects**: The Salesforce Administrator / Developer can create the additional objects inside the organization, to store the application specific data.

Ex: Branch__c, Customer__C, Policy__c, Agent__c, Position__C, Loan__C, Patient__c etc .

Custom Objects can be customizable and removable from the Organization.

All the Standard and Custom Objects will reside in the "Database.com".

2. Combination of both Standard and Custom Objects will be referred as "SObjects" (i.e., Salesforce Objects).

Governor Limits:
1. In Free Developer Edition: We can create maximum of 400 Custom Objects.

2. In Unlimited Edition: We can create maximum of 2,000 Custom Objects.

Ways to list out the Objects:
-
Salesforce provides the below 3 ways to list out the objects.

1. By using "Object Manager".
    Click on "Setup" menu.
        1. Click on "Object Manager" Tab.
        2. View the Salesforce Objects (Standard + Custom)
2. By using "Schema Builder":
    Click on "Setup" menu.
        1. Search for the option "Schema Builder" in Quick Find box.
        2. Go to the "Schema Builder" Tool.
        3. Go to the "Left Panel".
        4. Select the "Checkbox" for the required Object.
        5. View the Object Structure on the Canvas in Pictorial format.

3. By using "Developer Console".
    Click on "Setup" menu.
        1. Click on "Developer Console" link.
        2. Click on "File ---> Open".
        3. Select the entity Type as "Objects".
        4. View the Objects names in the Right Panel.

5. Double click on the "Object Name", to view the fields inside the object.

Ways to Create Custom Objects: Salesforce provides the below 2 ways to Create Custom Objects inside the Organization.

1. By using "Object Manager".
2. By using "Schema Builder".

While Creating a Custom Object, Salesforce provides the below 3 Layers by default.

Ex: Object Name: Branch.

1. Table / Model: Branch__c: Provides 5 Standard Fields / Columns.
    1. ID.
    2. Name
    3. Owner
    4. CreatedBy
    5. LastModifiedBy
2. Tab (User Interfaces): Branches
3. Business Logic (Class): Branch__c.

Navigation:

Click on "Setup" menu.

1. Click on "Object Manager" Tab.
2. Click on "Create ---> Custom Object".
3. Enter the Singular Label, Plural Label, and Description.
4. Go to the "Optional Features" section.
    1. Select the Checkbox "Allow Reports".
    2. Select the Checkbox "Allow Activities (Task, Event)".
    3. Select the Checkbox "Track Field History".
    4. Select the Checkbox "Allow in Chatter Groups".
5. Go to the "Deployment Status" section.
    1. In Development ---> Object is not ready to use.
    2. Deployed ---> Object is ready to use.
6. Go to the "Object Creation Options" section.
    1. Select the Checkbox "Add Notes & Attachments Related List".
    2. Select the Checkbox "Launch a New Custom Tab Wizard After Saving the Custom Object".
7. Click on "Save & Next" button.
8. Select the "Tab Style" to be applied for the Tab, by using "Lookup icon".
9. Click on "Next" button.
10. Make the Tab to be visible to all Profile users by selecting as "Default ON".
11. Click on "Next" button.
12. Make the Tab to be visible for All the Applications, by selecting the Checkbox "Include Tab".
13. Click on "Save" button.

Use Case: Create a Custom Object "Customer", to manage the Customers details. Create a Tab for the object to manage the records.

Use Case: Create a Custom object "Branch" to store all the branch details without providing the tab to manage the records.

Tabs:
Tab provides a collection of graphical user interfaces used to manage the records inside the Object.

By using tab, we can perform all the DML Operations (INSERT, UPDATE, DELETE, UNDELETE, VIEW) on the object records.
An Object can have only one tab, where object name should be in singular format and tab name should be in Plural format.
Salesforce provides the below 5 types of Tabs.

1. Custom Object Tabs.
2. Web Tabs.
3. Visualforce Page Tabs
4. Lightning Page Tabs
5. Lightning Component Tabs.

To create the tab for the Custom Object, we must select the "Custom Object Tabs" option.

Creating Custom Object Tab

Click on the "Setup" menu.

1. Search for the option "Tabs" in the Quick Find box.
2. Go to the "Custom Object Tabs" section.
3. Click on "New" button.
4. Select the Object Name from the Picklist, to which we need to create the tab.
5. Select the Tab Style, by using Lookup icon.
6. Click on "Next" button.
7. Make the Tab to be visible to All the Profile users, by selecting "Default ON".
8. Click on "Next" button.
9. Select the Checkbox "Include Tab", to make the tab to be visible in All Applications.
10. Click on "Save" button.

Observation: It will create a tab for the custom object and will represent the tab on the Navigation bar by default.

Fields: Field represents a column inside the table, which is used to store the Application / Business/Organization specific data.

Salesforce provides 2 types of fields.
1. Standard Fields: These are the readymade fields provided by Salesforce by default for each object.

Ex: Id, Name, Owner, CreatedBy, LastModifiedBy

We can Customize the Standard Fields up to some extent (i.e., we can change the label of the Field but, we cannot change the Field Datatype).
Standard Fields can't be removable from the Object.

2. Custom Fields: Based on the business requirement, we can add one or more additional fields to the object, to store the business specific data.

Ex: Email__c, Phone__c, Age__c, Gender__c, Title__c, Salary__C, Address__c etc.
Note: Custom Fields can be customized and removed from the object if they are no longer needed.

Governor Limits:
1. In Free Developer Edition: An object can have maximum of 500 custom fields.

2. In Unlimited Edition: An object can have maximum of 800 Custom fields.

Ways to list out the fields: We can view the fields inside the object in the following 3 ways.

1. By using "Object Manager".
2. By using "Schema Builder".
3. By using "Developer Console".

Datatypes: While creating a custom field, each field should be associated with a datatype, which describes the type of data the column can hold inside it.

Salesforce provides the below datatypes.

1. **Text**: This Datatype allows us to store maximum of 255 characters of content inside the field, which includes the alphanumerical values and special characters also.

Properties
1. Field Label: It is used to specify the label/Static text to be visible on the User Interface.

2. Field Name: It represents the actual column name inside the database object.

3. Description: It represents the description/Comments about the field.

4. Help Text: This property is used to specify the Tooltip Text message to be visible upon placing the mouse on the field.

5. Length: Used to restrict the number of characters to be allowed inside the field.

6. Required Checkbox: Used to make the field value required. (Mandatory).

Note: Salesforce provides the below 5 ways to make the field required.

1. By using "Required Checkbox".
2. By using "Page Layout Customizations".
3. By using "Validation Rules".

4. By using "Apex Triggers".
5. By using "Visualforce Programming".

7. Unique Checkbox: Used to make the field values unique.

8. External ID Checkbox: Used to store the external system Record Id inside the field, which can be used as the reference during integration.

2. **Text Area**: It allows us to store maximum of 255 Characters of content inside the field in multiple lines.

Properties:
1. Field Label
2. Field Name
3. Description
4. Help Text
5. Length
6. Required Checkbox:
7. Number of lines visible:

3. **Text Area Long**: This Datatype allows us to store maximum of 1,31,072 Characters of content inside the field in multiple lines.
Properties:
1. Field Label
2. Field Name
3. Description
4. Help Text
5. Length
6. Number of Lines Visible

4. **Text Area Rich**: This datatype allows us to store maximum of 1,31,072 characters of content in multiple lines along with the formatting options. i.e., we can change the font, style, size, color, insert images, hyperlinks, alignments, bullet formats etc.

Properties:
1. Field Label
2. Field Name
3. Description
4. Help Text
5. Length
6. Number of Lines Visible

5. **Checkbox**: It allows us to store the Boolean values inside the field. (i.e., TRUE / FALSE).
Properties:
1. Field Label
2. Field Name
3. Description:
4. Help Text:

5. Default Option (Unchecked)

6. **Phone**: This Datatype allows us to store the Phone / Fax values inside the field, which will be represented in US Phone Number format automatically by default.

Properties:
1. Field Label:
2. Field Name:
3. Description:
4. Help Text:
5. Required Checkbox:
6. Unique Checkbox:
7. External ID Checkbox:

7. **Number**: It allows us to store an 18-Digit Number, which includes the Decimal Digits.

Properties:
1. Field Label:
2. Field Name:
3. Description:
4. Help Text:
5. Required Checkbox:
6. Length: (Number of Digits to be Allowed)
7. Number of Decimal Digits:

8. **Percent**: This Datatype allows us to store the numerical values along with the decimal digits, by post-fixing the value with a "%" symbol.

Properties:
1. Field Label:
2. Field Name:
3. Description:
4. Help Text:
5. Length: (Max. Digits: 18)
6. Number of Decimal Digits:
7. Required Checkbox:

9. **Currency**: It allows us to store maximum of an 18-Digit number, which includes the decimal digits also. It represents the value by pre-fixing with the currently configured currency symbol inside the organization.
1. We can Change the Currency symbol based on the need from "Company Information" link.
2. We can Enable Multiple Currencies by using the Checkbox "Activate Multiple Currencies".

Properties:
1. Field Label:
2. Field Name:
3. Description:
4. Help Text:

5. Length:
6. Number of Decimal Digits:
7. Required Checkbox:
8. External Id Checkbox:

10. **Email**: It allows us to store the Email Id values inside the field. It contains the in-built email validation format.
Properties:
1. Field Label:
2. Field Name:
3. Description:
4. Help Text:
5. Required Checkbox:
6. Unique Checkbox:
7. External ID Checkbox:

11. **URL**: It allows us to store the Website Name / URL / Path of an Application inside the field.

Properties:
1. Field Label:
2. Field Name:
3. Description:
4. Help Text:
5. Required Checkbox:

12. **Date**: It allows us to store the Date values inside the field by selecting from the Pop-up Calendar in the form of "MM/DD/YYYY" format.

Properties:
1. Field Label:
2. Field Name:
3. Description:
4. Help Text:
5. Required Checkbox:

13. **Time**: This Datatype allows us to store the Time values in 12 / 24 Hours format inside the field.
Properties:
1. Field Label:
2. Field Name:
3. Description:
4. Help Text:
5. Required Checkbox:

14. **Date Time:** It allows us to store both Date and Time Stamp value inside a single field.

Ex: Created By, Last Modified By, Deletion Date...etc.

Properties:
1. Field Label:
2. Field Name:
3. Description:
4. Help Text:
5. Required Checkbox:

15. Picklist: Picklist Control allows us to populate a set of elements inside the control, from which the user can select only one element.
Properties:
1. Field Label:
2. Field Name :
3. Description:
4. Help Text :
5. Values : To specify the List of elements to be get populated inside the control.
6. Required Checkbox:
7. Checkbox : Arrange the Elements in Alphabetical Order.
8. Checkbox : Set the First Element as the Default Value.
9. Checkbox : Restrict the Picklist values based on the Values Set.

16**. Picklist-Multiselect:** This Control allows us to populate a group of elements inside the control, from which the user can select one or more elements.

Properties:
1. Field Label:
2. Field Name :
3. Description:
4. Help Text :
5. Values (LOV's):
6. Checkbox: Arrange the Elements in Alphabetical order.

17. **Text Encrypted:** This datatype allows us to store the content in encrypted format

Properties:
1. Field Label:
2. Field Name :
3. Description:
4. Help Text :
5. Length :
6. Required Checkbox:
7. Masked Type: (Ex: Password, Credit Card Number etc.)
8. Masked Character: *, X

18. **Geo Location:** It allows us to store the geographical coordinates of the specified location. (i.e., Latitude and Longitude values).

Properties:
1. Field Label:

2. Field Name:
3. Description:
4. Help Text:
5. Required Checkbox:

19. Auto Number: Auto Number Datatype allows us to make the field values to be get Auto Generated based on the specified format.
These are the Read-only fields whose value will be generated by Salesforce system automatically upon creating a new record inside the object.

System Generated field values cannot be editable by the Users.

Salesforce provides the below System Generated Fields.
Ex: Id, Created By, Last Modified By, Auto Number, Formula, Roll-up Summary Fields.

Properties:
1. Field Label:
2. Field Name:
3. Description:
4. Help Text:
5. Display Format: Product Code: P-{0000}
   Employee ID: EID-{000000}
6. Starting Number: Product Code: 1001
   Employee ID: 400001
7. Checkbox: Generate Auto Number for Existing Records.

Creating Fields: While creating the fields inside the object, we have the below 2 ways.

1. By using Object Manager.
2. By using Schema Builder.

Click on "Setup" menu.

1. Click on "Object Manager" Tab.
2. Click on the "Required Object" Name. (Ex: Patient)
3. Click on "Fields & Relationships" link from Left Panel.
4. Click on "New" button.
5. Select the "Datatype" from the List. (Ex: Email / Phone...)
6. Click on "Next" button.
7. Enter the Field Label, Field Name, and Description.
8. Select the Required Settings for the fields.
   (Ex: Required, Unique...etc.)
9. Click on "Next" button.
10. Select the "Visible" Checkbox, to make the fields to be visible to all profile users.
11. Click on "Next" button.
12. Click on "Save" button.

Observation: It will Add a new field to the object, and will represent the field on the user interface automatically.

Use Case: Create a Custom Object "Position", to store all the Open Jobs Details and add the below Custom Fields to the object.

1. Position Name : (Standard)
2. Location : Picklist (Mandatory)
   (LOV's: Hyderabad, Bangalore, Chennai, Pune, Noida, Delhi, Kochi)
3. Number of Vacancies: Number (Mandatory) (Max : 2)
4. Position Status : (Default : New Position)
   (LOV's: New Position, Open Approved, In Progress, Recruited, On Hold, Closed)
5. Open Date : Date
6. Milestone Date : Date (Mandatory)
7. Minimum Budget : Currency
8. Maximum Budget : Currency (Mandatory)
9. HR Contact Number : Phone (Mandatory)
10. HR Email ID : Email (Mandatory)
11. Travel Required : Checkbox
12. Passport Required : Checkbox
13. Position Description: Text Area Long.
14. Skills Required : Text Area Rich
15. Reference ID : Auto Number (PID-400001) PID-{000000}

Page Layout: Page Layout provides a collection of graphical user interfaces, which can be used to manage the records inside the object.

By using Page Layout, we can perform all the DML Operations on the object records. (Ex: Insert, Update, Delete, Undelete, View...etc)

While Creating a Custom Object, for each object Salesforce provides a Page Layout by default as below.

Object Name	**Page Layout Name**
Account	Account Layout
Contact	Contact Layout
Opportunity	Opportunity Layout
Case	Case Layout
Patient__c	Patient Layout
Customer__c	Customer Layout

Note: We can view the Page Layouts of an Object as below.

Click on "Setup" menu.

1. Click on "Object Manager" tab.
2. Click on the Required Object Name (Ex: Position)

3. Click on "Page Layouts" link from the left panel.
4. View the Page Layouts.

We can create additional Page Layouts for the object based on the business requirement.

Go to the "Page Layouts" link.

1. Click on "New" button.
2. Select the "Existing Page Layout" from the Picklist.
3. Enter the New Page Layout Name in the Textbox.
4. Click on "Save" button.

Note: Each Page Layout Contains 2 Modes as below.

1. Edit Mode / Edit Layout / Edit Page:
Edit Mode will represent the object fields in "Editable" format.
Edit Mode will get opened upon click on "New Button /Edit Button".

2. Detail Mode / Detail Layout / Detail Page:
Detail Mode will represent the detailed information of the record along with the related child records information.
Detail Mode will get opened upon Click on "Save" button.

Page Layout Customizations

We can customize the Page Layout based on the business requirement as below.

1. We can Add New Sections onto the Page layout.

2. We can specify the "Number of Columns", to be used to arrange the fields in each Section.

3. We can specify the "Tab-Key Order" for each Section.

4. We can place the required fields in each Section.

5. We can move the fields from One Section to another.

6. We can Make the Fields "Required / ReadOnly".

7. We can Remove Unnecessary fields from the Layout.

8. We can Add the required fields onto the Layout.

9. We can Remove Unnecessary Related Lists from the Layout.

10. We can Add the required Related Lists onto the Layout.

11. We can Customize the Fields to be Visible on the Related Lists.

12. We can Customize the Buttons to be visible on the Related Lists.

13. We can Arrange the Related Lists in the Required Order.

14. We can Add the Custom Buttons on to the Page Layout.

15. We can Embed the Visualforce Page onto the Layout.

16. We can Embed the Chart Component on to the Layout.

17. We can Embed the "Google Maps" onto the Layouts.

Navigation:

Click on "Setup" menu.

1. Click on "Object Manager" Tab.
2. Select the Required Object Name (Ex: Position)
3. Click on "Page Layouts" link from Left Panel.
4. Click on the "Page Layout" Name. (Ex: Position Layout)
5. Do the required Customizations inside the Page Layout Editor.
6. Click on "Save" button.

Schema Builder: Schema Builder is a graphical representational tool which allows us to perform the below 2 operations.

1. We can represent the existing objects structure on the Canvas in the Pictorial format.
2. We can build the schema based on the need. i.e., through Schema Builder, we can create our own Custom Objects, add the fields to the objects, and also map the relationship between the objects.

Use Case: Create a Custom Object "Hiring Manager", to store all the HR Peoples information inside it. Add the below fields to the object.

1. Hiring Manager Name : Standard Field.
2. Location : Picklist
   (LOV's: Hyderabad, Bangalore, Chennai, Pune, Mumbai, Noida, Delhi)
3. Designation : Picklist
   (LOV's: HR Executive, Sr. HR Executive, Recruitment Specialist, HR Manager)
4. Contact Number : Phone (Required)
5. Email Address : Email (Required)
6. Current CTC : Currency (Max: 8)

Drawbacks: 1. While creating a Custom Object through Schema Builder, it won't provide the tab for the object. (It will create the Object and will provide the Business Logic).

2. While adding the fields through Schema Builder, it will represent only the required fields on the Page Layout by default. It won't represent the optional fields on the layout.