Salesforce Lab Experiments:

(Professional Elective – 22IS633)

1. Salesforce Account and Playground Setup Exercise: Create salesforce account, Tailhead account to practice salesforce related challenges and the features with following requirements:

   → **Create your Salesforce Developer account** for accessing a dedicated Salesforce practice environment.

   → **Register for a Trailhead account**, Salesforce's official learning platform, to complete learning modules and challenges.

   → **Connect Developer account to Trailhead** to integrate learning progress with hands-on experimentation. Explore its features through identifying learning modules of your interest. Read Understand and experiment to accumulate your credit points on this platform.

   → **Set up and configure the Trailhead Playground**, enabling exploration of essential Salesforce features and configurations necessary for practical exercises and skill development.

2. Distinguish Standard objects and Custom objects in salesforce based on business requirements. Create a custom object for an organization based on use-case with following transaction criteria:
   An engineering college wants to track internship details of students on the Salesforce platform. Internship details consist company name, role offered, stipend, duration, and internship status. You will create a custom object for this purpose, add fields, and test record creation in your Trailhead Playground.
   - **Log in to Trailhead**:
     → Go to Trailhead site.
     → Log in with your Salesforce credentials.
   - **Launch Trailhead Playground**:
     → Click your profile icon → Hands-On Orgs → Launch the default playground.
   - **Navigate to Object Manager**:
     → Click Setup (Gear icon) → Setup.
     → Under Objects and Fields, click Object Manager.
   - **Create Custom Object**:
     → Click Create → Custom Object.
     → Enter Label: Internship
     → Plural Label: Internships
     → Object Name: Internship
     → Check Launch New Custom Tab Wizard after saving.
     → Click Save.

- **Create a Custom Tab:**
  → Select a tab style (e.g. briefcase icon).
  → Click **Next → Next → Save**.
- **Add Custom Fields to Internship Object**:

Field Label	Data Type	Details
Company Name	Text	80 character limit
Role Offered	Text	80 character limit
Stipend	Currency	2 decimal places
Duration (Months)	Number	Length: 2
Start Date	Date	DD/MM/YYYY
Status	Picklist	Values: Applied, Selected, Ongoing, Completed

  → Go to **Fields & Relationships → New** for each field.
  → Define field details as above and **Save**.
- **Test Record Creation:**
  → Go to **App Launcher (9 dots)** → search **Internships**.
  → Click **New** and enter sample data for a student internship.
  → Save the record.

3. On your trailhead playground with suitable settings create the record triggered flow. When an **Opportunity is marked as Closed Won**, the system should **automatically create a follow-up Task** assigned to the Opportunity Owner with: i) Subject: Follow up with client ii) Due Date: 7 days from today.

Creating Record-Triggered Flow for Create Records based on mentioned constraints required steps are as follows:

- **Navigate to Flows**
  → Go to **Setup > Flows**.
  → Click **New Flow**.
- **Select Flow Type**
  → Choose **Record-Triggered Flow**.
  → Click **Create**.
- **Configure Trigger**
  → **Object:** Opportunity
  → **Trigger the Flow When:** A record is updated.
- **Condition Requirements:** All Conditions Are Met (AND)
  →Field: Stage
  →Operator: Equals
  →Value: Closed Won
- **Optimize the Flow For:** Actions and Related Records.
  →**Add Create Records Element**
  →Click +, select **Create Records**.

→Label it: Create Follow-Up Task.

- **Configure Create Records**
  → **How Many Records to Create?** One
  → **How to Set the Record Fields?** Use separate resources and literal values.
  →**Object:** Task
- **Set Field Values:**
  → **OwnerId** → Opportunity OwnerId
  → **WhatId** → Opportunity Id *(links task to opportunity)*
  → **Subject** → *Follow up with client*
  → **ActivityDate** → Use formula to calculate Today + 7 days.
- **Add Formula Resource for Due Date**
  → Click **New Resource**.
  → **Resource Type:** Formula
  → **API Name:** DueDate7Days
  → **Data Type:** Date
  → **Formula:** TODAY() + 7
- **Assign Formula to ActivityDate**
  →Set **ActivityDate** = {!DueDate7Days}
- Save as **Auto-Create Follow-Up Task for Closed Won & Activate**

4. Using salesforce sales cloud activity into account, from the user's trailhead playground create a Flow that automates a task in creating a follow-up when a New Lead is created. Ensure Sales reps always follow up with new leads within 2 days of its creation.

We'll use **Record-Triggered Flow** in Salesforce Flow Builder.

Pre-Requisites

- You have **Leads** being created in Salesforce.
- Sales Reps are assigned to Leads via the **Owner** field or a related field.
- You want a **Task** to be automatically created with a **Due Date = Lead Created Date + 2 Days**

Step 1: Open Flow Builder in Trailhead Playground

→ Log in to your Trailhead Playground.
→ Click the **Gear Icon (Setup)** > **Setup**.
→ In the **Quick Find box**, type Flows.
→ Click **Flows**, then click **New Flow**.

Step 2: Select Flow Type

→ Choose **Record-Triggered Flow**.
→ Click **Create**.

Step 3: Configure Trigger as follows:

→ **Object: Lead**
→ Trigger the Flow When: **A record is created**
→ Condition Requirements: **None** (or you can filter e.g. Lead Status = Open)
→ Optimize the Flow For: **Actions and Related Records**

Step 4: Add Create Records Element

→ Now we add the action to create a task (follow-up).
→ Click the + icon below the Start element.
→ Choose **Create Records**.
→ Label it: Create Follow-Up Task.

Configure the Create Records Element:

- Label: **Create Follow-up Task**
- Object: Task
- Set Field Values:

→ Subject: Lead Follow Up
→ Status: Not Started
→ Priority: Normal
→ OwnerId: Lead.OwnerId *(Assign to the same rep who owns the Lead)*
→ WhatId: Leave blank (this is for opportunities/cases)
→ WhoId: Lead.Id *(this links the Task to the Lead)*
→ ActivityDate: TODAY() *(or {!$Flow.CurrentDate})*

- Save Flow with a clear label like: **Daily Lead Follow-Up &** Click **Activate**
- **Testing**: Create a new Lead and wait for a minute that triggers flow. Confirm that a Task is created and assigned to the Lead owner.

5. **Create a Screen Flow** in Salesforce that automates a trigger based on a commercial use-case consisting of collecting input via a screen and sends an email as an acknowledgment. The acknowledgement may be any of salesforce CRM formalities like opportunity, lead, or a case.

Example of one commercial Use Case**:** A Sales Rep manually logs a **Customer Meeting** via a **Screen Flow** attached to an Opportunity. After submission:

- The flow stores the meeting data. Creates a Task for the Sales Rep
- Sends an **email acknowledgment** to the customer confirming the meeting is logged.
- Optionally logs a Task for record-keeping.
- **Step 1: Add a Screen to Collect Meeting Info**

→ **Label:** "Log Customer Meeting"
→ Add the following input fields:

- o **Date** (Date Component): Meeting Date
- o **Text**: Meeting Location
- o **Long Text Area**: Meeting Notes & Optional: **Lookup Field** to choose Opportunity.

- **Step 2: Get Related Opportunity Contact**

→ Element: **Get Records**
→ Label: Get Opportunity
→ Object: Opportunity
→ Filter: Id Equals {!OpportunityId} *(from screen input or passed as variable)*
→ Store first record.

Now get the Contact element records

→ Element: **Get Records**
→ Object: OpportunityContactRole
→ Filter: OpportunityId Equals {!Opportunity.Id} AND IsPrimary Equals True
→ Store first record.

Save the Contact’s Email from this record (Contact.Email) for use in the email ste

- **Step 3: Create a Task for Internal Tracking (Optional)**
  - → Element: **Create Records**
  - → Object: Task
  - → Field values:
    - o Subject: "Customer Meeting Logged"
    - o WhoId: {!Contact.Id}
    - o WhatId: {!Opportunity.Id}
    - o ActivityDate: {!MeetingDate}
    - o Description: {!MeetingNotes}
    - o OwnerId: {!Opportunity.OwnerId}
- **Step 4: Send Email Acknowledgement**
  - → Use **Action** → Type: Send Email
  - → Label: Send Meeting Confirmation
  - → Email fields:
    - o **To Address**: {!Contact.Email}
    - o **Subject**: “Thank you for meeting with us!”
    - o **Body**: Contextual / Conventional / suitable message context

    [**Tip:** You must enable **Send Email action** in **Process Automation Settings**: Setup → Process Automation Settings → Enable "Send Email Action"]
- **Step 5: Add Finish Screen**
  - → Add a final screen saying “Meeting logged and email sent.”
  - → Save and Activate Flow:
  - → Label: Log Customer Meeting and Send Acknowledgement
  - → Save → **Activate**
  - → Add Flow to Lightning Page or Button **-** Add it as a **button** on Opportunity page.
  - → Or embed it in the **Opportunity** Lightning Pag**e** using the **Flow component**.
  - ✓ **Test**: Sales reps can log customer meetings with a few clicks. The flow automatically:
  - → Records the Task internally.
  - → Sends a personalized email to the customer.
  - → Improves CRM hygiene and customer communication.

6. **A)**For a Project Management application used at project consultant, has a custom object Project__c has fields : Start_Date_c (Date field), End_Date_c (Date field) along with other fields like Client_name_c, Project Title_c, Project_Domain_c etc. relevant to the business needs. Create a Validation Rule for data input on Project_c object to prevent user input errors, where End Date is earlier than Start Date.preventing while saving a record.

Step 1. Navigate to Object Manager
→ Go to Setup.
→ Click on Object Manager.
→ Search and select Project__c *(custom object)*.
Step 2. Open Validation Rules
→ In the left panel, click Validation Rules.
→ Click New.

Step 3. Define the Validation Rule
Rule Name:
EndDate_After_StartDate
Description:
Ensures that End Date is after Start Date in Project records.

Step 4. Write the Error Condition Formula:

```
AND(
   NOT(ISBLANK(Start_Date__c)),
   NOT(ISBLANK(End_Date__c)),
   End_Date__c < Start_Date__c
)
```

Step 5. Define Error Message
In Error Message box, enter:
"End Date must be later than Start Date."

Step 6. Select Error Location
→ Field: Choose End_Date_c so the error message appears near End Date field.

B) In an organization MIS, when creating or editing a Contact object record, the Email field must be mandatory only if the Preferred Contact Method (another custom field) is set to Email to ensure data accuracy by enforcing Email entry conditionally, not universally. Create a ‘Validation Rule’ on Contact object that throws an error if, Preferred Contact Method = 'Email' and Email field is blank.

- Navigate to Object Manager, Select Contact.
- In the left panel, click Validation Rules. Click New.
- Define the Validation Rule:
  Rule Name: Email_Required_When_Preferred_Email
- Error Condition Formula:
  CopyEdit

  ```
  AND(
     ISPICKVAL(Preferred_Contact_Method__c, "Email"),
     ISBLANK(Email) )
  ```

- Define Error Message: In Error Message box, enter the following-
  "Email address is required when Preferred Contact Method is Email.
- Select Error Location: Select Email so the error shows near the field itself..
- Click Save to activate

7. In an organization’s Opportunity object, if the Stage is 'Closed Won', then the Amount must be greater than zero, and Close Date must not be in the past. Create ‘validation rule’ that ensures closed deals have a valid monetary value and cannot be marked closed in the past accidentally.

Objective in this solution to the given problem is :

- **Create a Validation Rule** on Opportunity to enforce:

→ Stage = Closed Won, and

→ Amount ≤ 0, or Close Date < Today → Throw error.

- **Step 1-** In the trailhead playground navigate to Object Manager
- **Step 2- Define Validation Rule:**

→ In the left panel, click **Validation Rules**. > Click **New**.

→ **Rule Name:** ClosedWon_ValidAmount_CloseDate

- **Step 3-Write the Error Condition Formula**

```
AND(
   ISPICKVAL(StageName, "Closed Won"),
   OR(
    Amount <= 0,
CloseDate < TODAY()    ))
```

- **Step 4- Define Error Message,** In Error Message box, enter:"For Closed Won Opportunities, Amount must be greater than zero and Close Date cannot be in the past."
- **Step 5- Select Error Location Field**: Choose Stage or leave at Top of Page as it relates to multiple fields. Click Save to activate the validation rule.

8. The Sales Manager wants to analyze monthly opportunities by stage and display them visually for team meetings. Create a 'Matrix Report' showing Opportunity Amounts grouped by Stage and Close Month, and build a 'Dashboard' component to display this data into a chart for management review.

- **Step 1. Create a New Report**

→ Go to **App Launcher → Reports**.

→ Click **New Report**.

→ Choose **Opportunities** as the report type.

→ Click **Continue**.

- **Step 2. Configure Report Format**
  - In the Report Builder, click on **Outline**.
  - Drag **Stage** to the **Rows** field.
  - Drag **Close Date** to the **Columns** field.
  - Group **Close Date** by **Calendar Month** (click the dropdown beside Close Date rouping and select 'Group Date By: Calendar Month').
- **Step 3. Summarize Data, Save and Run**

→ Click the dropdown on **Amount** field and choose **Summarize → Sum** to display total opportunity amounts.

→ Click Save & Run.

→ Name it Monthly Opportunity by Stage Report, select a folder, and save.

- Step 5. Create a Dashboard

→ Go to Dashboards from App Launcher.

→ Click New Dashboard.
→ Enter Sales Performance Dashboard as name and select folder.
→ Click Create.

- Step 6. Add Component

→ Click + Component.
→ Select the Monthly Opportunity by Stage Report created earlier.
→ Choose a suitable chart type, e.g., Stacked Bar Chart for visualizing Stage vs. Month.
→ Click Add, Click Save and Done to finalize the dashboard

9. With help of a specific business case, create a report as summary of CRM formality containing statistics of business opportunities won / lost, stage of business process and monetary involvements in those processes.

Business-Scenario: The Sales Manager wants to analyze Opportunities closing this month where:

- Where Stage is Negotiation/Review or Proposal/Price Quote
- Amount is greater than Rs.1,00,000
- Grouped by Opportunity Owner, showing sum of Amount for each owner to plan targeted follow-ups and forecast revenue.

- **Step 1: Navigate to Reports**

→ Go to App Launcher.
→ Search and click Reports.
→ Click New Report button (top right).

- **Step 2: Select Report Type**

→ In the Choose Report Type window:

- Select Opportunities.
- Click Continue.

- **Step 3. Apply Filters**

On the report builder:

i. Close Date:
   - Find Close Date filter under Filters pane.
   - Set it to Current Month.

ii. Stage filter:
   - Click Add Filter.
   - Field: Stage
   - Operator: equals
   - Value: select Negotiation/Review and Proposal/Price Quote (multi-select).

iii. Amount filter:
   - Click Add Filter.
   - Field: Amount
   - Operator: greater than
   - Value: 100000

- **Step 4. Group Rows by Opportunity Owner**

→ In the Outline pane (left side), under GROUP ROWS, click Add group.

→ Select Opportunity Owner.
→ The report will now show Opportunities grouped by each owner.

- **Step 5. Summarize Amount Field**

→ In the Columns pane, find the Amount column.
→ Click the dropdown next to Amount.
→ Click Summarize, then choose Sum.
→ This will show total Opportunity Amount per owner.

- **Step 7. Adjust Columns**

→ Keep necessary columns like: Opportunity, Name, Stage, Amount, Close Date, Opportunity Owner.
→ Remove any irrelevant columns to make the report clear.

- **Step 8. Save and Run Report**

→ Click Save & Run (top right).
→ Enter the following details:

  - Report Name: Opportunities Closing This Month Over Rs.1,00,000 /-
  - Folder: Save in an appropriate folder (e.g., Sales Reports).

10. Sales team of an Electronics component mfg. & exports company is lacking the real-time visibility into key performance metrics like lead conversion rate, opportunity pipeline, sales revenue, and individual rep performance. With help of Salesforce advanced feature of data visualization create a Dash Board for any suitable use-case of your choice.

To design and implement a Salesforce Dashboard that provides actionable insights into sales performance, enabling data-driven decisions and enhanced visibility across the sales organization.

- **Goal:** Improve visibility into sales performance.
- **Key Metrics to include:** Lead conversion rate, Opportunity stage breakdown, Closed-won revenue (monthly/quarterly), Sales rep performance, Average deal size, Sales pipeline value, Top opportunities

- **Step 1: Define Dashboard Goals and Metrics**
- **Step 2: Determine Data Sources**

→ Use Salesforce **standard and custom objects**:

  - Leads, Opportunities, Accounts, Activities (Tasks, Events)
  - Custom fields (if applicable, e.g., industry, region)

- **Step 3: Create Custom Report Types (if needed)**

→ Navigate to Setup > Report Types
→ Create custom report types to include specific relationships (e.g., Opportunities with Activities)
→ Ensure necessary fields are exposed for reporting

- **Step 4: Build Source Reports**

→ Go to Reports > New Report
→ Create **summary reports** for each KPI:

  - Opportunities by Stage
  - Leads by Source and Conversion Status
  - Closed Opportunities by Owner and Amount
  - Pipeline by Forecast Category

→ Use filters to segment data (time period, region, product line, etc.)

- **Step 5: Create the Dashboard**
  - → Go to **Dashboards > New Dashboard**
  - → Select appropriate layout (2-column, 3-column, etc.)
  - → Add components for each metric:
    - o **Bar chart**: Sales by Rep
    - o **Donut chart**: Opportunity Stage Distribution
    - o **Gauge**: Monthly Revenue Goal Progress
    - o **Table**: Top 10 Opportunities
    - o **Line chart**: Revenue Trend over Time
- **Step 6: Apply Filters (Dynamic Dashboards)**
  - → Add filters like:
    - o Time period (e.g., last 30 days, quarter, YTD)
    - o Region or Territory
    - o Product or Service line
  - → Consider **Dynamic Dashboards** (viewer sees data according to their own access)
- **Step 7: Test and Validate**
  - → Check data accuracy against real records
  - → Review with stakeholders to confirm dashboard meets needs
  - → Adjust charts or filters based on feedback
- **Step 8: Schedule and Share**
  - → Schedule automatic email delivery (daily/weekly/monthly)
  - → Set appropriate access and visibility rules
  - → Train users on how to interpret the dashboard
- **Step 9: Monitor and Improve**
  - → Collect feedback after 2–4 weeks
  - → Adjust reports or visuals as business needs evolve
  - → Add new components as more metrics become relevant

***