**Internship Report**

A Salesforce Developer Virtual Internship Report Submitted to

**Jawaharlal Nehru Technological University Anantapur,**

**Anantapuramu**

in partial fulfillment of the requirements for the award of the degree of

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND SYSTEMS ENGINEERING**

*Submitted by*

**PULLAIAHGARI JAGRUTHI 19121A1587**



Department of Computer Science and Systems Engineering

**SREE VIDYANIKETHAN ENGINEERING COLLEGE**

(AUTONOMOUS)

(Affiliated to JNTUA, Ananthapuramu, Approved by AICTE, Accredited by NBA & NAAC) Sree Sainath Nagar, Tirupati – 517 102,A.P

INDIA

2021-2022

Department of Computer Science and Systems Engineering

**SREE VIDYANIKETHAN ENGINEERING COLLEGE**

**(AUTONOMOUS)**

(Affiliated to JNTUA, Ananthapuramu, Approved by AICTE, Accredited by NBA & NAAC) Sree Sainath Nagar, Tirupati – 517 102, A.P., INDIA



**Certificate**

This is to certify that, the Salesforce Virtual Internship entitled

**“Salesforce Developer Virtual Internship”**

is the bonafide work done by

**PULLAIAHGARI JAGRUTHI 19121A1587**

in the Department of C**omputer Science and Systems Engineering**, **Sree Vidyanikethan Engineering College (Autonomous), Sree Sainath Nagar, Tirupati** and is submitted to **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** for partial fulfillment of the requirements of the award of B.Tech degree in Information Technology during the academic year 2021-20

**Certificate**



**ABSTRACT**

Salesforce platform provides a simple and straightforward way to build applications for various purposes. In this internship, the goal was to Build a Data Model for a Travel Approval App and use Formulas and Validations to implement automated mechanisms in the application. A data model provides a structured way to store data within the application being built. Whereas, A validation rule can contain a formula or expression that evaluates the data in one or more fields and returns a value of either True or False, depending on which actions can be taken on how to process the data within the application.

Through salesforce developer catalyst we can Meet the tools and technologies that power development on the Salesforce platform, get introduced to the platform, navigate use cases, and build custom functionality, learn about Salesforce multi-tenancy architecture and create a sample application and also earn two superbadges: apex specialist and process automation specialist. Salesforce delivers breakthrough productivity for all users because it puts the customer—employees, partners, consumers, and devices. Salesforce has introduced artificial intelligence (AI) on its Einstein platform, helping to improve the analytics process. Developers at Salesforce developer training evaluate business processes and develop CRM workflows, and custom solutions for different business needs.

**TABLE OF CONTENTS**

**[Lightning App Builder] 1**

1. Lightning App Builder-Agenda 2
   1. Get Started with the Lightning App Builder 2

Meet the Lightning App Builder 2

How the Lightning App Builder Works 2

Lightning Components 3

* 1. Build a Custom Home Page for Lightning Experience 4

Create a Custom Home Page for Lightning Experience 4

* 1. Build a Custom Record Page for Salesforce Mobile App 6

Create a Custom Lightning Record Page 7

Create an activity tab. 8

Make Your Record Page Dynamic 9

* 1. Build an App Home Lightning Page 10

The App Home Lightning Page 10

Create an App Page 11

Test Your App Page in Lightning Experience 12

* 1. Work with Custom Lightning Components 13

Install a Custom Lightning Component 13

Add the Custom Lightning Component to Your App Page 13

Test the Custom Component 13

### [Flow Builder] 15

1. Flow Builder - Agenda 16
   1. Learn About Flow Resources and Variables 16

Flow Resources 16

Introducing Variables 17

* 1. Create a Variable 18

Prepare to Create a Variable 18

|  |  |  |
| --- | --- | --- |
| Create a Variable |  | 20 |
| Store Information in a Flow Variable |  | 20 |
| 2.3 Add Screens to Your Flow |  | 21 |
| Introducing Screen Components |  | 21 |
| The Runtime Experiences |  | 22 |
| The Screen Element |  | 23 |
| Name Component |  | 24 |
| Add a Confirmation Screen |  | 25 |
| 2.4 Add Logic to your Flow |  | 26 |
| Create Branches with Decision Elements |  | 26 |
| Update the Flow |  | 27 |
| Create a New Decision |  | 27 |
| Reconnect the Elements |  | 28 |
| 2.5 Add Actions to Your Flow |  | 29 |
| Get Records |  | 29 |
| Update Records Elements |  | 29 |
| Create Records Elements |  | 30 |
| **[Help Article: Automate Business Processes]** |  | **39** |
| 3. Help Article - Automate Business Processes |  | 40 |
| **[Automate Business Process for a Recruiting App** |  | **42** |
| 4. Automate Business Process for Recruiting App | - Agenda | 43 |

* 1. Build a Process for creating Interviewer Records 43

[Creating a process with an immediate action 43](#_TOC_250053)

[Test the Process 44](#_TOC_250052)

* 1. [Lay the Ground for an Approval Process 45](#_TOC_250051)

[Create a Folder and Email Templates 45](#_TOC_250050)

[Create Fields 47](#_TOC_250049)

* 1. [Create an Approval Process 47](#_TOC_250048)

[creating your multistep approval process. 48](#_TOC_250047)

[Set up email alerts. 51](#_TOC_250046)

* 1. [Create a Process for Submitting Positions for Approval 51](#_TOC_250045)

Use Process Builder to Create a New Process 51

* 1. [Create a Candidate Rating Flow 52](#_TOC_250044)

[Create Radio Buttons for Experience 52](#_TOC_250043)

[Deploy the Flow 55](#_TOC_250042)

[Build a Discount Approval Process] 58

1. Discount Approval Process - Agenda 59
   1. Prepare Your Org 59

Add a New User 59

Create a Role 59

* 1. Create an Approval Process 60
  2. Create Initial Submission Actions 62
  3. Create a Process for Submitting Positions for Approval 64
  4. Create a Candidate Rating Flow 65

Create Radio Buttons for Experience 66

Add Screen Elements 68

Deploy the Flow 69

[[Quick Start: Lightning App Builder] 70](#_TOC_250041)

1. [Quick Start: Lightning App Builder - Agenda 71](#_TOC_250040)
   1. [Create Your First Page 71](#_TOC_250039)

[Introduction 71](#_TOC_250038)

[Create Your First Page 71](#_TOC_250037)

* 1. [Add More Components 73](#_TOC_250036)

[Add Components to the Page 73](#_TOC_250035)

* 1. [Add Quick Actions and Activate the App 74](#_TOC_250034)

[Add Quick Actions and Configure the Page 74](#_TOC_250033)

[Check Out Your New App 75](#_TOC_250032)

[[Quick Start: Process Builder] 76](#_TOC_250031)

1. [Quick Start: Process Builder - Agenda 77](#_TOC_250030)
   1. [Create a New Process on the Account Object 77](#_TOC_250029)

[Introduction 77](#_TOC_250028)

[Create a New Process on the Account Object 77](#_TOC_250027)

* 1. [Add Process Criteria 78](#_TOC_250026)

[Create Criteria 78](#_TOC_250025)

* 1. [Add Your Process Action 79](#_TOC_250024)

[Create an Action 79](#_TOC_250023)

* 1. [Test Your Process 80](#_TOC_250022)

[Check the Process 80](#_TOC_250021)

[[Data Management] 82](#_TOC_250020)

1. [Data Management - Agenda 83](#_TOC_250019)
   1. [Import Data 83](#_TOC_250018)

[Introduction to Data Import 83](#_TOC_250017)

[Prepare for Data Import 84](#_TOC_250016)

[Use the Data Import Wizard 85](#_TOC_250015)

* 1. [Export Data 87](#_TOC_250014)

[Introduction to Data Export 87](#_TOC_250013)

[Using the Data Export Service 88](#_TOC_250012)

[[Event Monitoring] 89](#_TOC_250011)

1. [Event Monitoring - Agenda 90](#_TOC_250010)
   1. [Getting Started with Event Monitoring 90](#_TOC_250009)

[What is Event Monitoring ? 90](#_TOC_250008)

[A Quick Note About the API 93](#_TOC_250007)

* 1. [Query Event Log Files 94](#_TOC_250006)

[View Events in Salesforce Event Log File Browser 94](#_TOC_250005)

[Query Event Log Files in Developer Console 95](#_TOC_250004)

[Event Log File Documentation for Event Types 96](#_TOC_250003)

* 1. [Download and Visualize Event Log Files 98](#_TOC_250002)

[Download Event Log Files Using Python 98](#_TOC_250001)

[Visualize Event Log File Data 98](#_TOC_250000)

|  |  |  |
| --- | --- | --- |
| **List Of Figures** | | |
| **S.N O 1** | **Figure name**  Fig 1.1: App builder components page | **Page No** 3 |
| **2** | Fig 1.2: Activating new home page | 6 |
| **3** | Fig 1.3: Activity tabs | 9 |
| **4** | Fig 1.4.1: App home lightning page | 10 |
| **5** | Fig 1.4.2: Lightning experience tab | 12 |
| **6** | Fig 1.5: Custom components page | 14 |
| **7** | Fig 2.1: Create variables | 26 |
| **8** | Fig 2.3.1: Runtime Experience | 31 |
| **9** | Fig 2.3.2: New screen elements | 32 |
| **10** | Fig 2.3.3: Name Components | 33 |
| **11** | Fig 2.4.1: Branch creation and elements | 39 |
| **12** | Fig 2.4.2: Reconnect the elements | 41 |
| **13** | Fig 2.5.1: Get Records on your Flow | 40 |
| **14** | Fig 2.5.2: New Action Element | 45 |
| **15** | Fig 2.5.3: Connect the elements | 47 |
| **16** | Fig 5.5.1: Edit screen flow label | 55 |
| **17** | Fig 5.5.2: Selecting components | 59 |
| **18** | Fig 5.5.3: Deploy the flow | 60 |
| **19** | Fig 6.1: Page setup and save | 62 |
| **20** | Fig 6.2: Add more components | 65 |
| **21** | Fig 7.2: Add process criteria | 67 |
| **22** | Fig 7.3: Add Actions to the Process | 69 |
| **23** | Fig 9.1.1: Event monitoring analysis | 81 |
| **24** | Fig 9.2.1: Event log files | 83 |
| **25** | Fig 9.2.2: Event log files running page | 85 |
| **26** | Fig 9.2.3: Report event type fields | 86 |
| **27** | Fig 9.3 : Lightning Components | 88 |

**MODULE-1**

**[Lightning App Builder]**

1. **Lightning App Builder-Agenda**
   1. [Get Started with the Lightning App Builder](https://trailhead.salesforce.com/content/learn/modules/lightning_app_builder/lightning_app_builder_intro?trailmix_creator_id=trailblazerconnect&trailmix_slug=salesforce-developer-catalyst)
   2. [Build a Custom Home Page for Lightning Experience](https://trailhead.salesforce.com/content/learn/modules/lightning_app_builder/lightning_app_builder_homepage?trailmix_creator_id=trailblazerconnect&trailmix_slug=salesforce-developer-catalyst)
   3. [Build a Custom Record Page for Lightning Experience](https://trailhead.salesforce.com/content/learn/modules/lightning_app_builder/lightning_app_builder_recordpage?trailmix_creator_id=trailblazerconnect&trailmix_slug=salesforce-developer-catalyst).
   4. [Build an App Home Lightning Page](https://trailhead.salesforce.com/content/learn/modules/lightning_app_builder/lightning_app_builder_apphome?trailmix_creator_id=trailblazerconnect&trailmix_slug=salesforce-developer-catalyst)
   5. [Work with Custom Lightning Components](https://trailhead.salesforce.com/content/learn/modules/lightning_app_builder/lightning_app_builder_custom_comps?trailmix_creator_id=trailblazerconnect&trailmix_slug=salesforce-developer-catalyst)
   6. **Get Started with the Lightning App Builder**

Learning Objectives

After completing this unit, you’ll be able to:

* Understand how the Lightning App Builder can help you build responsive apps and custom pages for Lightning Experience and the mobile app
* Understand the layout of the Lightning App Builder user interface
* Describe the difference between a Lightning page and a Lightning component

**Meet the Lightning App Builder**

Your users are busy. They’re closing deals, providing top-notch service, and marketing to your prospects and customers. By creating customized pages, you can put key information at your users’ fingertips and provide them with an easy interface to update and add records.

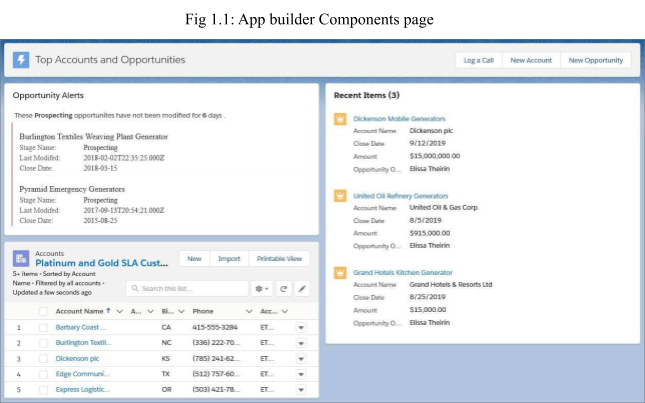
The Lightning App Builder is a point-and-click tool that makes it easy to create custom pages for the Salesforce mobile app and Lightning Experience, giving your users what they need all in one place.

But that’s not all. When you edit a Lightning app from the App Manager in Setup, you’re brought into the Lightning App Builder to manage the app’s settings. You can update the app’s branding, navigation, options, and manage the Lightning pages assigned to that app all in the Lightning App Builder.

**How the Lightning App Builder Works**

With the Lightning App Builder, you can build:

* Single-page apps that drill down into standard pages
* Dashboard-style apps, such as apps to track top sales prospects or key leads for the quarter
* “Point” apps to solve a particular task, such as an expense app for users to enter expenses and monitor expenses they’ve submitted
* Custom record pages for your objects, tailored to the needs of your users
* Custom Home pages containing the components and features that your users use most



**Lightning Components**

A Lightning component is a compact, configurable, and reusable element that you can add to a Lightning page in the Lightning App Builder.

Lightning pages support these components:

* **Standard Components** - Standard components are Lightning components built by Salesforce.
* **Custom Components** - Custom components are Lightning components that you or someone else have created. You can configure custom Lightning components to work in Lightning App Builder.
* **Third-Party Components on AppExchange** - The AppExchange provides a marketplace for Lightning components. You can find packages containing components already configured and ready to use in the Lightning App Builder.
  1. [**Build a Custom Home Page for Lightning Experience**](https://trailhead.salesforce.com/content/learn/modules/lightning_app_builder/lightning_app_builder_homepage?trailmix_creator_id=trailblazerconnect&trailmix_slug=salesforce-developer-catalyst)

Learning Objectives

After completing this unit, you’ll be able to:

* Create a custom Home page for your Lightning Experience users
* Assign different Home pages to different profiles
* Set the default home page

●

**Create a Custom Home Page for Lightning Experience**

We’ll tweak the position of the components on the standard home page layout slightly to give you an idea of what’s possible.

Let’s get started!

1. From Setup, enter App Builder in the Quick Find box, then click Lightning App Builder.
2. Click New, select Home Page, then click Next.
3. Step through the wizard and name the page New Home Page, select the Standard Home Page template, and then click Finish.

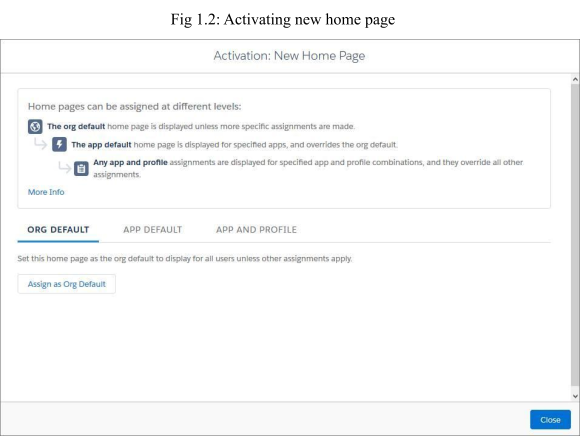
An empty home page opens, ready for you to build it. The components pane contains all the standard components available for a home page, plus any custom components you have installed in your org.

1. Drag the Assistant component to the top-right region.
2. Drag Performance to the top-left region.
3. Add Today’s Events to the lower left region and add Key Deals to the lower right region.
4. Add Today’s Tasks above the Assistant.
5. Click Save.
6. But wait, what’s this? There’s more? Yes, there is. Saving the page isn’t enough to get it out to your users. Lightning pages must be activated before your users can see them. Normally, if you aren’t done with your page, or aren’t ready to make it public, you can click Not Yet here to save the page and return to the App Builder. But that’s not us. We’re bold! We’re done with our page and want to give it to our users right now!
7. Click Activate, and we’ll do just that.

If you saved previously and didn’t activate the page, you can click the Activation button in the toolbar to be ready for the next section.

Roll Out Your Custom Home Page to Your Lightning Experience Users

When activating a home page, you have three different options: You can make your page the default for everyone in the org, the default for an app, or assign it to one or more app and profile combinations, giving your users access to a page designed just for their role.



Let’s assign this home page to the System Administrator profile so we can go look at it afterward.

* + Click App and Profile, then click Assign to Apps and Profiles.
  + Select the Sales app, then click Next.
  + Scroll down the list of profiles and select System Administrator, then click Next.
  + Review the assignment, and then click Save.

That’s it. Now all users with the System Administrator profile see your New Home Page while working in the Sales app. Let’s go have a look

* 1. **Build a Custom Record Page for Salesforce Mobile App**

Learning Objectives

After completing this unit, you’ll be able to:

* + Create a customized object record page for Lightning Experience and the Salesforce mobile app.
  + Add visibility rules to a record page component.
  + Activate the custom record page for your users.

**Create a Custom Lightning Record Page**

Let’s build a custom opportunity record page.

We’ll tweak the standard record page layout just a bit, so you can get a feel for how things go together. After you’re comfortable with that, you can go to town and customize your record pages any way you like. Let’s get started.

From Setup, enter App Builder in the Quick Find box, then select **Lightning App Builder**.

* + Click **New**.
  + Select **Record Page** and start stepping through the wizard.
  + Name your page New Opportunity Page, and select **Opportunity**.

Start typing an object’s name in the Object field to filter the list and find what you’re looking for more quickly.

* + Choose the **Header, Subheader, Right Sidebar** template, and click **Finish**.

In the components pane, you see all the standard components available for opportunity record pages and any custom components that you’ve installed in your org.

* + From the settings menu ( ), select **Always show icons**.

The icons that appear in the palette show what form factors each component supports. For example, if you add the Chatter Feed component to your page, it displays when you view the page on both a desktop and in the Salesforce mobile app. The same is not true for Chatter Publisher, which is supported on desktop only. We’ll see this behaviour in action when we test our finished page.

Drag the Highlights Panel component into the top region of the page. Click **See How It Works** in the component properties pane to find out where the highlights panel content comes from.

* + Add the Path component to the region below the highlights panel.
  + Add a Chatter component to the lower right region.
  + Add a Tabs component to the lower left region
  + The Tabs component comes with some default tabs already in place. Let’s add more.
  + In the Tabs component details pane, click **Add Tab**.

By default, another Details tab is added. But because we already have one, let’s change this new one to something else.

* + Click the second **Details** tab.

From the Tab Label dropdown menu, select **Custom**, and give the tab a new label: Recent Items

Click **Done**.

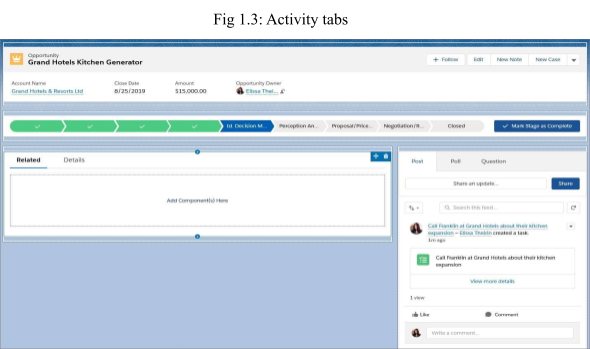
**Create an activity tab.**

Drag the Recent Items tab to the top of the Tabs list in the properties pane.

The Recent Items tab is now in the first position in the tab’s component. You can click around between the tabs, but nothing changes because the tabs don’t have any components in them. They’re empty. Let’s fix that.

* + Select the Details tab.
  + Drag a Record Detail component right below the Details tab, into the green highlighted area
  + Add a Related Lists component to the Related tab, an Activities component to the Activity tab, and the Recent Items component to the Recent Items tab.
  + Select the tabs component on the canvas, and in the properties pane, change the order of the tabs to: Details, Activity, Recent Items, then Related.

Click **Save**, then **Not Yet**.



**Make Your Record Page Dynamic**

Component visibility properties appear when you select a component on a record, app, or home page in the Lightning App Builder. This behaviour applies to standard components, custom components, and components from AppExchange. No need to do anything to your custom components. It’s all handled by the Lightning App Builder. If you don’t define a filter, the component displays on the Lightning page as usual. When you define one or more filters and set the filter logic for a component, the component is hidden until the filter logic criteria are met.

Add a Rich Text component above the Chatter component on the page.

Enter this text in the component: A million-dollar opportunity closed! Oh yeah!

In the component properties, make the text bold and centered, 18-point size, and change the font to something that appeals to you.

Keep **Display as card** selected.

This setting makes the text inside the component more readable on Lightning pages by adding a white background to it instead of a transparent one. Toggle the setting off and back on to see what we mean.

* + Click **Add Filter**.
  + Set Field to **Amount**, if it’s not already.
  + Set Operator to **Greater Than or Equal**. For Value, enter 1000000.
  1. **Build an App Home Lightning Page**

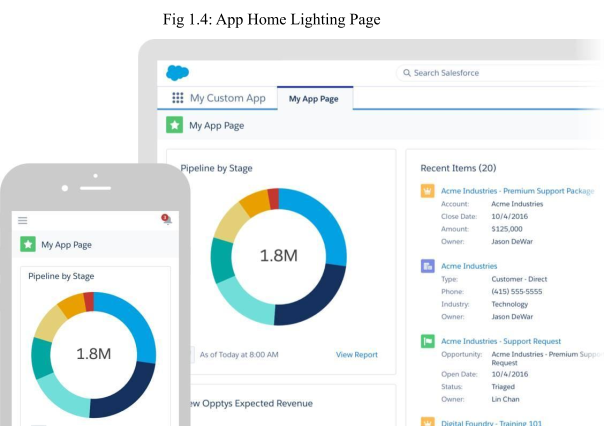
Learning Objectives

After completing this unit, you’ll be able to:

* + Add components to a Lightning page
  + Configure the properties of a Lightning page and a Lightning component
  + Add actions to a Lightning page
  + Add a Lightning page to Lightning Experience and the mobile app

**The App Home Lightning Page**

Add a custom home page for an app to the Salesforce mobile app and Lightning Experience app navigation bars to let your users easily access the objects and items that are most important in that app.



**Create an App Page**

Let’s build an app home page for a sales team.

Your sales team needs to see top deals in the pipeline, with a visual interface that makes it easy to absorb key details at a glance. They want to see the most recent opportunities they’ve viewed and be able to drill into the record details with a single tap or click. And they want functionality to log calls and create accounts and opportunities on the go.

Let’s get started!

* + From Setup, enter App Builder in the Quick Find box, then select **Lightning App Builder**.
  + Click **New**.
  + Select **App Page**, and then click **Next**.
  + Name your Lightning page Top Accounts and Opportunities.
  + Select the **Two Regions** template, and click **Finish**.

If the Lightning App Builder walkthrough pops up, dismiss it.

* + Drag the List View component into the first region.

In the properties pane, select **Account** for the object, select the **Platinum and Gold SLA Customers** filter, and set the number of records to display to 5.

Add a Recent Items component into the second region.

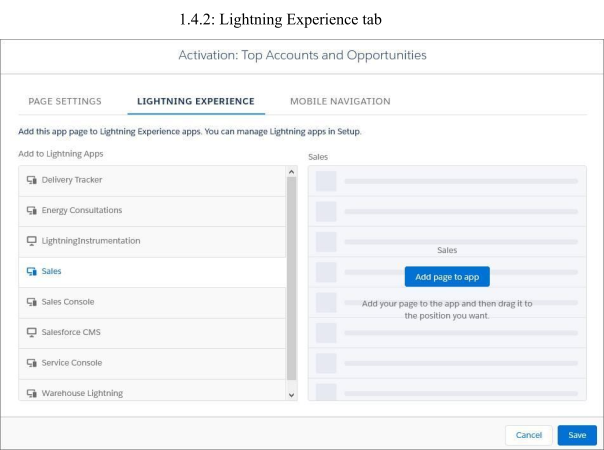
In the properties pane for the Recent Items component, click **Select** and configure the Selected list so it only contains Opportunity.

**Add Your App Page to Lightning Experience and the Salesforce Mobile App**

Just like the other pages, your users can’t access your app page until you activate it. During activation, you can customize the page’s custom tab label, adjust its visibility, and set its position in the Salesforce mobile app and Lightning Experience app navigation bars, all in one place.

Click the **Lightning Experience** tab.

Select the Sales app, and click **Add page to app**.



* + Click the **Mobile Navigation** tab.
  + Select **Mobile Navigation Menu**, and click **Add page to app**.

This adds the app page to the mobile navigation in the “Mobile Only” Lightning app in the Salesforce mobile app. Adding the page to the Sales app in Lightning Experience like we did previously, ensures that the app page appears in the Sales app within the Salesforce mobile app as well.

By default, new pages you add to the navigation in the “Mobile Only” app appear below the Smart Search Items menu item. If you leave the Top Accounts and Opportunities page there, it will appear in the Apps section of the menu. We don’t want that, so let’s move it up.

Drag the page to below the Today menu item.

**Test Your App Page in Lightning Experience**

You’ve created your page and activated it. Now let’s see it in action! If you’re still in the App Builder, click **Back** to return to Setup.

From the App Launcher ( ), find and select **Sales**.

Click **Top Accounts and Opportunities** from the app navigation bar.

Here’s your Lightning page! The lightning bolt icon appears here too, and the three actions that you added are in the highlights panel.

* 1. **Work with Custom Lightning Components**

Learning Objectives

After completing this unit, you’ll be able to:

* + Install a custom Lightning component
  + Use your mobile device to preview your app in the Salesforce mobile app

**Install a Custom Lightning Component**

We’ve provided a custom Opportunity Alert Lightning component that you can add to your Lightning page. Let’s install it into your org.

Launch your Trailhead Playground by going to any unit with a hands-on challenge, scrolling to the bottom of the page, and clicking **Launch**. If you see a tab in your org labeled Install a Package, great! Follow the steps below.

* + Click the **Install a Package** tab.
  + Paste 04t2E00000161fSQAQ into the field.
  + Click **Install**.
  + Select **Install for Admins Only**, then click **Install**.

After the installation completes, it’s time to add the component to your page.

**Add the Custom Lightning Component to Your App Page**

From Setup, enter App Builder into the Quick Find box, and then click **Lightning App Builder**.

* + Click **Edit** next to the Top Accounts and Opportunities page that you created in the previous unit.
  + Add the Opportunity Alert component above the List View component.

**Test the Custom Component**

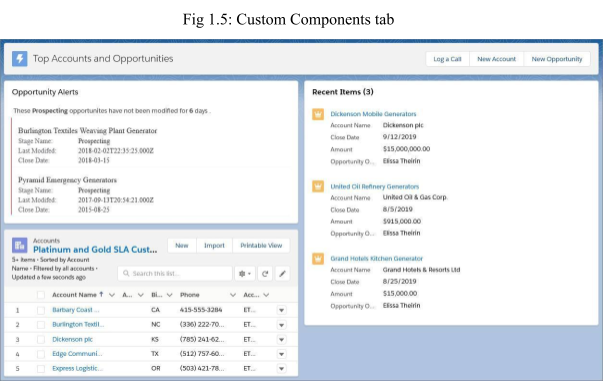
Let’s look at it in the Salesforce mobile app first.

* + Open the app on your mobile device.
  + If necessary, log in using your Trailhead Playground credentials.
  + Go to the Top Accounts and Opportunities page in the Sales app and scroll down until you see the new component.

You might have to refresh the screen to tablet see your changes.

Go back to your org and view the Top Accounts and Opportunities page in Lightning Experience.

Because you created the page using the Two Columns template, it uses the two-column format when you view it on the desktop.



Congratulations! You used the Lightning App Builder to create an app page, personalize your Lightning Experience Home page, and customize a Lightning Experience record page.

**MODULE-2**

**[Flow Builder]**

1. **Flow Builder - Agenda**
   1. Learn About Flow Resources and Variables
   2. Create a Variable
   3. Add Screens to Your Flow
   4. Add Logic to your Flow
   5. Add Actions to Your Flow
   6. **Learn About Flow Resources and Variables**

Learning Objectives

After completing this unit, you'll be able to:

* + - List the resources available in Flow Builder.
    - Describe what a flow variable is.

**Flow Resources**

In flows, resources are placeholders similar to merge fields in an email template or a formula. Let's say you start an email with Hi, {!$User.FirstName}.

{!$User.FirstName} is a placeholder, so when the email is sent, it displays the actual first name of the user. In each step of the flow (the elements added to the canvas), you can reference flow resources instead of manually entering values.

Let's go over the basic kinds of flow resources available in Flow Builder.

* + - **Constant** represents a fixed value, such as a tax rate.
    - **Choice, Picklist Choice Set, or Record Choice Set** represents an option in a screen component. With the choice resource, you explicitly set each option's label and value. Choice sets, on the other hand, generate choices for you by using a filtered list of records or the values of a picklist (or multi-select picklist) field in your org.

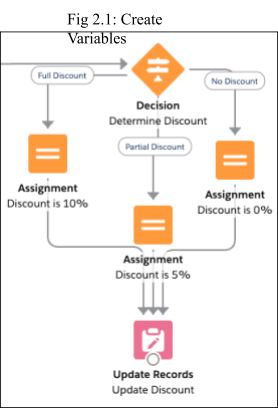
Example: In the New Contact flow, the screen prompts the user to select the associated account. Rather than building individual options for each account in your org, a record choice set generates the options. To generate an option for each account record in your org, the record choice set uses the default filters. Every choice consists of two components: a label to display in the screen component, and a value to use when the choice is

referenced elsewhere in the flow. The way this record choice set is configured, the screen component displays the account name for each option.

* + - Formula represents a calculated value, similar to a formula field. For example, create a formula that calculates 30 days from today, then reference that formula to set an opportunity close date.
    - Text Template represents some formatted text. For example, format the body of an email or Chatter post in a text template, then reference the text template in the appropriate action.
    - Variable represents a value that can change throughout the flow. We're going to spend the rest of this unit diving deep on variables.

**Introducing Variables**

The word variables can make you think of algebra classes or writing code in some scary language like Apex. However, they're an important thing to understand when you're building flows. Don't worry, no programming (or math) experience is required.



* 1. **Create a Variable**

Learning Objectives

After completing this unit, you'll be able to:

* + - Create a flow variable.
    - Define input and output variables.

**Prepare to Create a Variable**

Before you create your variable, figure out what kind of variable you need.

**What type of data should the variable store?**

First consider what type of data the variable will store, much like you do when you create a custom field.

|  |  |
| --- | --- |
| **Data Type** | **Use to store...** |
| Text | IDs, descriptions, and other text or long text data. |
| Record | Field values from a record, such as an opportunity. When this data type is selected, you create a record variable. |
| Number | A numeric value. |
| Currency | A currency value. |
| Boolean | A yes/no value, such as whether a checkbox was selected. |
| Date | A date value. |
| Date/Time | A date and time value. |
| Picklist | A picklist value. |
| Multi-Select Picklist | Picklist values, separated by semicolons. |

|  |  |
| --- | --- |
| Apex-Defined | Field values from an Apex class. When this data type is selected, you create an Apex-defined variable. |

**How many values should the variable store?**

By default, variables store one value that's compatible with the selected data type. If the data type is Number, the variable stores one numeric value. If the data type is Record, the variable stores field values for one record.

But what if you need to store multiple values in one variable, such as multiple email addresses? When you create a variable, you can enable it to do just that with the Allow multiple values checkbox. When that option is selected, you create a collection variable.

**Should the value be available from outside the flow?**

Each variable in your flow gets its value from somewhere.

Some variables get their values from inside the flow, such as when the user enters something in a screen component or the flow looks up field values from a record.

Other variables get their values from outside of the flow—namely, from whatever started the flow. For example, if you distribute a flow on an Account record page, you can pass the account's ID into the flow by using the flow's input variables. An input variable is a variable that has the Available for input checkbox selected.

**Where do you plan to use the variable?**

If you already know where you plan to use the variable, review documentation to see whether that field requires a certain kind of variable.

Tip: Pay special attention to which data types the field supports and whether it supports single values or collection values.

For example, you need to automatically submit an opportunity for approval from a flow. Rather than using the default approval settings, you need the approval request to be sent to a specific user, so you need to set the Next Approver Ids field. When you look at the reference documentation for the

Submit for Approval core action, you see that the Next Approver Ids field only accepts Text collection variables. So you need to store the ID of that user in a Text collection variable.

**Create a Variable**

1. From Setup enter Flows in the Quick Find box and click Flows.
2. Click New Flow.
3. Select Screen Flow and click Create.
4. From the toolbox, click Manager.
5. Click New Resource.
6. For Resource Type, select Variable.
7. Enter an API name and description for your variable.
8. Select the appropriate data type.
9. If you want to store multiple values (a collection), select Allow multiple values.
10. For record variables, select the object whose record values you plan to store.
11. Identify the variable's availability outside the flow.

**Store Information in a Flow Variable**

Once you've created a variable, you've got a placeholder just waiting for a value to… well, hold. A variable is only as good as the values stored in it, and Flow Builder offers a few ways to populate that variable. Let's review them.

|  |  |
| --- | --- |
| **How to Populate a Variable** | **Description** |
| Set the variable's default value | If you know what the initial value of the variable should be, set its default value when you create it. The variable keeps that value until it gets overwritten by another element. |
| Get a value from a record and put that value in the variable | When you configure a Get Records element (which we cover later in this module) or a record choice set, specify the fields to store and which variables to store |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | them in. |
| Change element | the | variable's | value | from | an | The most obvious way to change a variable's value is with an Assignment element. But other elements let you store values in variables. For example, the Create Records element lets you store the created record's ID in a variable. |
| Pass a value in from outside the flow | | | | | | Depending on how you distribute your flow, you can pass a value into an input variable. For example, if you add the flow to a Contact record page, you can pass the contact's ID into a flow variable. |

Now that we've got variables under our belt, let's dig in to the element categories available in Flow Builder. First up: screens.

* 1. **Add Screens to Your Flow**

Learning Objectives

After completing this unit, you'll be able to:

* + - List the types of components you can add to screens.
    - Add a confirmation screen to a flow.

**Introducing Screen Components**

Just as you configure the user experience of your record pages in Lightning App Builder, you use screens to do the same for your flow users.

Each screen is made up of one or more screen components. A screen component is a configurable, reusable element added to a screen.

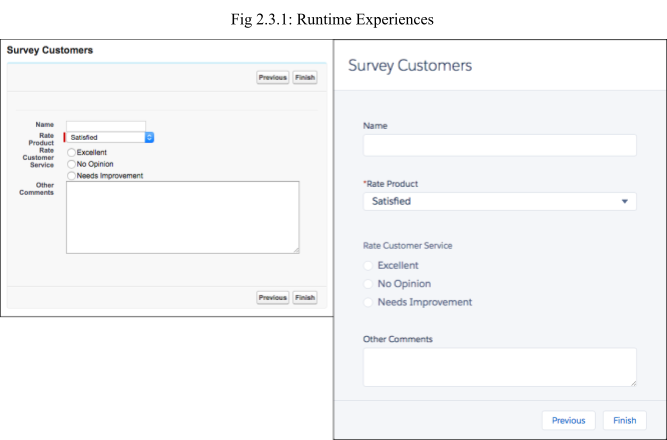
Screen components are available in three categories.

* + - Input includes standard components that request information from the user.
    - Display includes standard components that display information to the user.
    - Custom includes components that you or someone else have created. Install them from AppExchange or a third-party library, or work with a developer to build your own.

**The Runtime Experiences**

There's one more important consideration for screen components: which flow runtime experience they're supported in.

Flows have two different runtime experiences: Lightning runtime and Classic runtime. Like its name suggests, Lightning runtime looks and feels like Lightning Experience, while Classic runtime looks and feels like Visualforce.



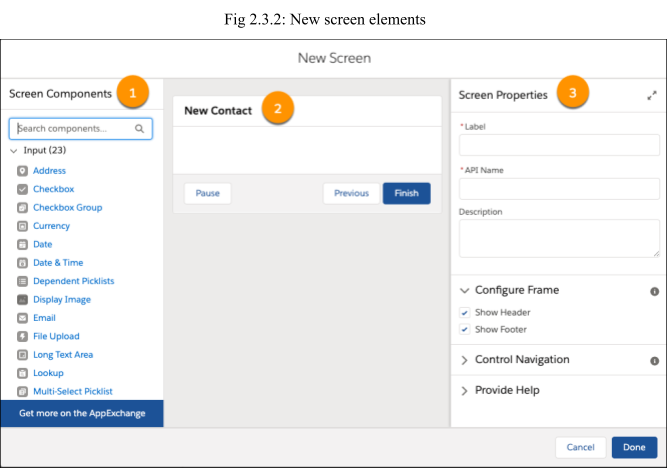
That said, the two runtime experiences aren't tied to either desktop experience. You can use Lightning runtime in Salesforce Classic, and you can use Classic runtime in Lightning Experience.

All screen components are supported in Lightning runtime, but not all screen components are supported in Classic runtime. Here are three indicators that a component requires Lightning runtime.

* + - The component icon is a lightning bolt.
    - No preview is available for the component.
    - A warning that a screen component requires Lightning runtime appears when the flow is saved.

**The Screen Element**

Let's break down the Screen element.



* + - Screen Components Pane (1): The left-side pane displays all the screen components available in your org. Click and drag a component to add it to the screen. Tip: Use the search field to easily find the screen component you need.
    - Screen Canvas (2): The canvas is where you build your screen. Drag components to arrange them in the right order.
    - Properties Pane (3): Depending on the canvas selection, the properties pane shows either the screen's properties or the properties of the selected component. To view or modify the screen properties, click the header or

footer in the canvas.

**What You Did in the Project**

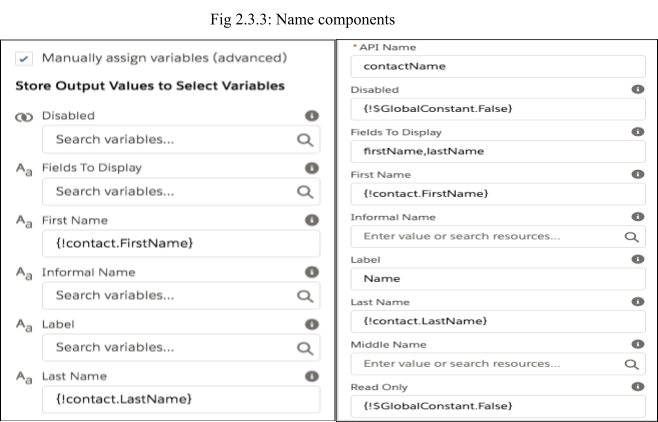
Let's look at the screen you built in the [Build a Simple Flow](https://trailhead.salesforce.com/content/learn/projects/build-a-simple-flow) project. In the New Contact flow, open the screen.

1. From Setup, enter Flows in the Quick Find box, and select Flows.
2. Open the New Contact flow.
3. From the canvas, double-click Contact Info.

The screen contains three components that request information about the contact: the name, the associated account, and a toggle that determines what to do when a contact with that name already exists.

**Name Component**

First, let's look at the Name component.



* + - The component isn't disabled, nor is it read-only.
    - The component displays fields for only First Name and Last Name. (The Name component can also display other name fields like Middle Name.)

If your project's Name component displayed a field for Salutation, the options would be Mr., Mrs., and Ms.

* + - The label for the component is Name.

You need to be able to reference the user-entered values for First Name and Last Name in other parts of the flow. Enter the Store Output Values section.

The First Name and Last Name values are stored in fields on the {!contact} record variable.

**Add a Confirmation Screen**

1. If you haven't already, open the "New Contact" flow that you created in the [Build a Simple Flow](https://trailhead.salesforce.com/content/learn/projects/build-a-simple-flow) project. Ensure the Freeform option is selected.
2. Drag a Screen element onto the canvas.
3. Give the screen a label and confirm.
4. Scroll or tab to the Configure Footer section, then select the Hide Previous radio button. Leave the other screen properties as is.
5. Add a Display Text component to the screen. From the screen components pane, search for Text and drag Display Text onto the canvas.
6. Give the Display Text component an API name: confirmation\_message. Now let's craft a message that thanks the user and confirms what the flow did. This flow has multiple branches. It either creates a new contact or updates an existing one. Ideally the confirmation messages is either, “Thanks! The contact was created.” or “Thanks! The contact was updated.” To provide custom confirmation messages, you can:
7. Create one static confirmation message that works for all possibilities. For example, Thanks! The contact was created or updated. This option is easy, although the user will immediately wonder, Well… which one?
8. Create a dynamic confirmation message that changes depending on the outcome of the flow.
9. Create a separate confirmation screen for each possibility. (To keep our flow lean, let's leave this as the last resort.)
10. Because only one word changes between the two messages, all it takes to make a dynamic message work is a simple formula.
11. In the text box, enter Thanks! The contact was XYZ. (Don't worry, XYZ is a placeholder for the formula.)
12. Now, let's figure out what the formula should be
    1. **Add Logic to your Flow**

Learning Objectives

After completing this unit, you'll be able to:

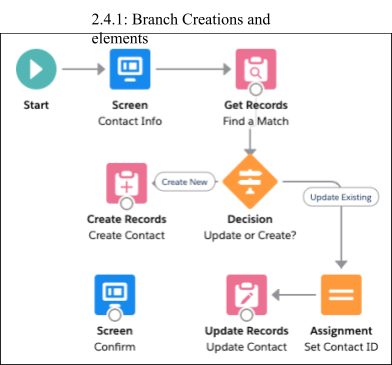
* + - List the logic elements available in Flow Builder.
    - Add branching logic to a flow.
    - Change a variable value in a flow.

**Create Branches with Decision Elements**

For most business processes, rules determine whether a particular action should be taken or not. Maybe a case should be escalated when the associated account is at risk. The rule in play here is “the associated account is at risk.” In Flow Builder, you evaluate the rules of your business process by using Decision elements.

Each decision answers a question, and the outcomes provide the possible answers. The answers can be as simple as Yes and No, but there's no limit to the number of answers. You determine which answer is appropriate by configuring conditions for each outcome. When a flow executes a Decision element, it evaluates each decision outcome in order. If an outcome's conditions are met, the flow takes the associated connector and ignores the other outcomes. If an outcome's conditions are not met, it evaluates the next outcome in the list.

Flow Builder provides the last outcome for you, which has no conditions: the default outcome. It acts as a fallback. When the conditions for every other outcome aren't met, the flow takes the connector for the default outcome.



As an example, let's look at the decision you built in the New Contact flow.

**Update the Flow**

The way the flow is currently built, it looks for a matching contact record before determining whether to update the existing one or create a new one. That's a wasted data element: Why look up the ID of a matching record if the user opts to create a new contact anyway?

**Create a New Decision**

1. Drag a Decision element onto the canvas.
2. Give the decision a label: Update If Existing? The API Name is automatically set to Update\_If\_Existing.
3. Give the first outcome the label Yes. Update the API Name to Update\_Yes. This way, the label for the decision connector is easy to read, but you can easily differentiate the name of that outcome from others in the Manager tab.
4. Leave When to Execute Outcome as is, and configure one condition for the outcome.
5. Under Outcome Order, click Default Outcome. Change its label to No. If you leave the outcome label as Default Outcome, it might not be obvious

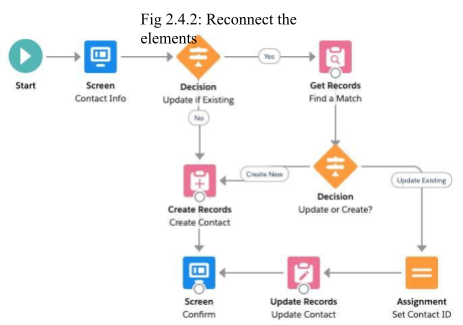
when that outcome is executed.

1. Click Done.

**Reconnect the Elements**

You're almost done! To finish, you need to make sure the Update If Existing? decision is executed before the Find a Match element.

1. Remove the connector between Contact Info and Find a Match.
2. Click the connector. When it's highlighted, the color changes to blue.
3. Press the Delete key.
4. Connect Contact Info to Update If Existing?.
5. Connect Update If Existing? to Find a Match. When prompted, select the Yes outcome, and click Done.
6. Connect Update If Existing? to Create Contact. Since there's only one outcome left to connect (No), Flow Builder automatically selects it for you.
7. Save the flow, and ignore the FYI warning.



Now the flow has two decision points. First, it evaluates whether to look for a matching record, based on whether the user opted to always create a new contact or not. Second, it evaluates whether the Find a Match element found a matching contact or not

* 1. **Add Actions to Your Flow**

Learning Objectives

After completing this unit, you'll be able to:

* + - List the actions available in Flow Builder.
    - Understand when to use a record variable in a data element.
    - Configure a Post to Chatter core action.

**Get Records**

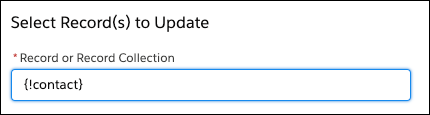
In a Get Records element, you identify which object to look up, how to filter the object's records, and how to sort the filtered records. Then you specify how many records to store and whether you want to manually store variables.

**Update Records Elements**

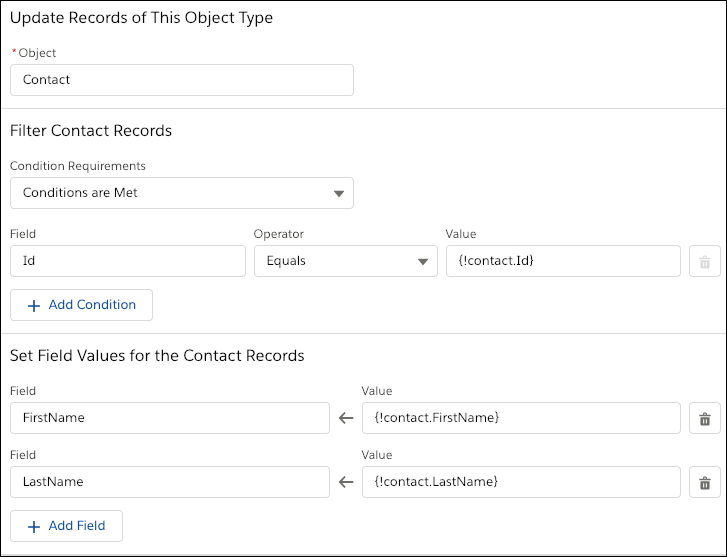
In an Update Records element, you identify which records to update and how to update those records.

* + - If you choose to use a record variable or record collection variable, the flow uses the IDs in that variable to identify the records to update, then updates the records by using the other field values in the variable.
    - Otherwise, you can manually set filter conditions to identify the records to update. With this option, you also manually identify the field values to change.

Here's a comparison of what the Update Contact element in the New Contact flow looks like for each option.



Record variable option



Manual option

**Create Records Elements**

To create multiple records, you must use the values from a record collection variable. Earlier in the flow, populate the record collection variable with the new records' field values. For each record that's created, the flow stores the ID of the created record in the ID fields of the record collection variable.

To create one record, you have two options for setting the record's values.

* + - If you've populated a record variable with the field values for the new record, choose to set the record fields by using all the values from a record variable. Then select the record variable or record collection variable to use. When the record is created, the flow stores the ID of the created record in the record variable's ID field.
    - To manually map values from various resources in the flow, choose to set the record fields by using separate variables, resources, and literal values. The ID of the created record is automatically stored for you. Optionally, you can manually store the ID of the created record in a Text variable, by selecting Manually assign variables (advanced).

**MODULE-3**

**[Help Article: Automate Business Processes]**

1. **Help Article - Automate Business Processes**

Instead of relying on your users to perform each part of a business process, automate it! The benefits are two-fold: your users can spend their time on other tasks, and you can trust that the process is always done just so. Salesforce offers tools to automate several kinds of business processes: guided visual experiences, behind-the-scenes automation, and approval automation. You'll be happy to know that these tools don't require you to write code—they're all point-and-click.

* + **Which Automation Tool Do I Use?** Salesforce provides a robust set of automation features to help you save time and resources. Use Flow Builder to automate most of your organization’s repetitive business processes. More features can provide further automation functionality, including approval processes, Flow Orchestration, Einstein Next Best Action, and Apex.
  + **Migration to Flow** You have used Workflow Rules or Process Builder to create a triggered process. Now, Flow Builder is the optimal tool. Why? It combines the capabilities of Workflow Rules and Process Builder in a single point-and-click tool. Use Flow Builder to automate if/then business processes and see a graphical representation of your automation as you build it.
  + **Flow Builder** Whether they’re buying movie tickets, paying bills, or changing restaurant reservations, when a customer interacts with a company, they expect a seamless and personalized experience. Flow Builder, formerly Salesforce Flow, provides declarative process automation for every Salesforce app, experience, and portal with point-and-click automation.
  + **Flow Orchestration** As your company grows, so does the complexity of your workflows. Processes often require input from multiple users in multiple departments across multiple time zones. This increased complexity results in an increased amount of time spent waiting for each person to complete their task in the proper order.
  + **Einstein Next Best Action** Display the right recommendations to the right people at the right time with Einstein Next Best Action. Create and display offers and actions for your users that are tailored to meet your

unique criteria. Develop a strategy that applies your business logic to refine those recommendations. Your strategy distills your recommendations into a few key suggestions, like a repair, a discount, or an add-on service. Display the final recommendations in your Lightning app or Experience Builder site.

* + **OmniStudio**

OmniStudio provides a suite of services, components, and data model objects that combine to create Industry Cloud applications. Create guided interactions using data from your Salesforce org and external sources.

* + **OmniStudio for Vlocity** OmniStudio for Vlocity is a set of services, components, and data model objects used to create Vlocity Industry Cloud Apps.
  + **Automated Actions** An automated action is a reusable component that performs some sort of action behind the scenes—like updating a field or sending an email. After you create an automated action, add it to a process, milestone, or other automated process.
  + **Approval Processes** It’s likely that you’re familiar with process automation in the form of workflow rules. Approval processes take automation one step further, letting you specify a sequence of steps that are required to approve a record.
  + **Workflow Rules** Workflow rules let you automate standard internal procedures and processes to save time across your org. A workflow rule is the main container for a set of workflow instructions. These instructions can always be summed up in an if/then statement.
  + **Process Builder** Many of the tasks you assign, the emails you send, and other record updates are vital parts of your standard processes. Instead of doing this repetitive work manually, you can configure flows or processes to do it automatically. We strongly recommend using Flow Builder, but Process Builder can also help you automate your business processes and give you a graphical representation as you build it.

**MODULE-4**

**[Automate Business Process for a Recruiting App]**

1. **Automate Business Process for Recruiting App - Agenda**
   1. Build a Process for creating Interviewer Records
   2. Lay the Ground for an Approval Process
   3. Create an Approval Process
   4. Create a Process for Submitting Positions for Approval
   5. Create a Candidate Rating Flow
   6. **Build a Process for creating Interviewer Records**

**Set Up a Process with Process Builder**

The first thing to tackle is automated interviewer records. To set this up, we use the Lightning Process Builder, which automates tasks using conditions and actions. With Process Builder, we define evaluation criteria, determining which objects the rule we create applies to. establish the criteria that records must meet to trigger the rule. Then you set up time triggers that determine when the process action fires.

Once your criteria are in place, immediate or time-dependent actions are set in motion. These actions can include creating a record, updating a record, sending an email, posting to Chatter, initiating a quick action, submitting for approval, launching a flow, or calling an Apex class.

### Creating a process with an immediate action

1. From Setup, enter Process Builder in the Quick Find box and select Process Builder.
2. Click New, then click Continue with Process Builder.
   * Name: Create Interviewer Record
   * The process starts when: A record changes
3. Click Save.
4. Click + Add Object.
5. Select Position from the Object picklist.
6. Under Start the process, select only when a record is created.
7. Click Save.
8. Click + Add Criteria.
9. For Criteria Name enter Hiring Manager Not Blank.
10. Select Conditions are met under Criteria for Executing Actions, then set the

conditions.

* + Field: Position: Hiring\_Manager c
  + Operator: Does Not Equal
  + Type: Global Constant
  + Value: $GlobalConstant.Null

1. Click Save.
2. Under Immediate Actions, click + Add Action and fill in the details.
   * Action Type: Create a Record
   * Action Name: Create Interviewer Record
   * Record Type: Interviewer
3. Under Set Field Values, relate the Employee field to the Hiring Manager.
   * Field: Employee
   * Type: Field Reference
   * Value: Hiring\_Manager c
   * Click Choose
4. Relate the interviewer record to the position record by clicking + Add Rowand filling in the details.
   * Field: Position
   * Type: Field Reference
   * Value: Record ID
5. Click Save, Activate, then Confirm.
6. Click Back To Setup.

### Test the Process

Now that the process is activated, we can test the process you just created.

But first, add a hiring manager user.

1. From Setup, enter Users in the Quick Find box and select Users.
2. Click New Userand fill in the details.

|  |  |
| --- | --- |
| **Field** | **Value** |
| First Name | Kathy |
| Last Name | Cooper |

|  |  |
| --- | --- |
| Email | enter your own email |
| Username | kcooper@[your initials + your favourite color].com (e.g., kcooper@APyellow.com) |
| Nickname | kcoop |
| Title | Customer Support Manager |
| Role | Customer Support, North America |
| User License | Salesforce Platform |
| Profile | Standard Platform User |

1. Deselect Generate new password and notify user immediately.
2. Click Save.

Now test the process you created.

1. Click App Launcher to open the App Launcher, then click Recruiting.
2. Click Positions, then New.
3. Select the record type Nontechnical Position, then click Next and fill in the details.
4. Click Save.
5. Click the Related tab.
6. From the Interviewers related list, click an Interviewer number and note the values in the Employee and Position fields.

## Lay the Ground for an Approval Process

After completing the set up the approval process in the next step we have to create a folder and Email Templates.

### Create a Folder and Email Templates

Here Start by creating a couple of email templates and a folder to house them.

1. From Setup, enter Classic Email Templates in the Quick Find box, then select Classic Email Templates.
2. Click Create New Folder.
3. Fill in the folder information:
   * Email Template Folder Label: Position Request Responses
   * Public Folder access: Read/Write
4. Ensure This folder is accessible by all users is selected.
5. Click Save.

Now create the necessary templates. First, set up a Position Approved email template.

1. Click New Template.
2. Ensure Text is selected, then click Next and fill in the details.

|  |  |
| --- | --- |
| **Field** | **Value** |
| Folder | Position Request Responses |
| Available for Use | Select |
| Email Template Name | Position Approved |
| Subject | Your position request was approved |
| Email Body | Dear {!Position c.OwnerFirstName}, Good news! Your recent position request has been approved.  Please log in to your org for details. |

1. Click Save.

Next, set up a Position Rejected email template.

1. Navigate back to the initial Classic Email Templates page, then click New Template.
2. Ensure Text is selected, then click Next and fill in the details.

|  |  |
| --- | --- |
| **Field** | **Value** |
| Folder | Position Request Responses |

|  |  |
| --- | --- |
| Available for Use | Select |
| Email Template Name | Position Rejected |
| Subject | Your position request was rejected |
| Email Body | Dear {!Position c.OwnerFirstName},  Unfortunately, your recent position request has been rejected.  Please log in to your org for details. |

1. Click Save.

### Create Fields

Now build a few fields on the Position object to use in the approval process.

1. From Setup, click Object Manager and select Position.
2. Click Fields & Relationships, then click New.
3. Choose Lookup Relationship as the Data Type, and click Next.
4. Select User from the Related To picklist and click Next.
5. Enter Approver 1 as the Field Label.
6. Click Next, Next, then Save & New.
7. Repeat steps 3 through 6 using Approver 2 as the Field Label.
8. Repeat steps 3 through 5 using Approver 3 as the Field Label.
9. Click Next, Next, and Save.

With the prep work finished, you can get to work setting up the approval process for the recruiting tea

## Create an Approval Process

Set Up a Multistep Approval Process

Our org is prepared, so you’re ready to set up the approval process for positions. An approval process is an automated process that approves records in Salesforce.When you build an approval process, you specify the steps necessary for approval. A given step can apply to all records or just records with certain attributes. You also specify who does the approving at each step and the actions to take when a record is approved, rejected, or recalled.

### creating your multistep approval process.

1. From Setup, enter Approval Processes in the Quick Find box and select Approval Processes.
2. Select Position from the Manage Approval Processes For picklist.
3. Click Create New Approval Process and select Use Standard Setup Wizard from the picklist, then fill in the details.
4. For Process Name, enter New Position Approval.
5. For Description, enter: All positions must be approved by the recruiter's manager and two to three other approvers. These approvers are tracked in approver fields on the Position object.
6. Click Next
7. At the picklist for Use this approval process if the following, select criteria are met.
8. Click Next.
9. In the Next Automated Approver Determined By picklist, select Manager.
10. Select Use Approver Field of Position Owner.
11. Select Administrators OR the currently assigned approver can edit records during the approval process.
12. Click Next, then Next.

Select fields to display on the approval page layout.

1. Select these fields from the Available Fields list.
   * Department
   * Education
   * Job Description
   * Location
   * Pay Grade
   * Salary Range
   * Skills Required
2. Click Add to add them to the Selected Fields list.
3. Select Display approval history information in addition to the fields selected above.
4. Select Allow approvers to access the approval page only from within the Salesforce application (Recommended).
5. Click Next.
6. As Submitter Type, select Owner from the Search picklist.
7. Ensure Position Owner is visible in the Allowed Submitters box.
8. Select Allow submitters to recall approval requests.
9. Click Save.
10. Select I'll do this later. Take me to the approval detail page to review what I've just created.
11. Click Go!

**Create Approval Steps**

You’ve created the basic approval process. Next, create initial submission actions.

1. Under the Initial Submission Actions related list, click Add New, then select Field Update.
2. For Name, enter Approval Status to Pending.
3. For Field to Update, select Approval Status from the picklist.
4. Under Specify New Field Value, select A specific value, then select Pending.
5. Click Save.

**Now create approval steps.**

1. Click New Approval Step under the Approval Steps related list.
2. For Name, enter Manager of Position Owner.
3. Click Next.
4. Select All records should enter this step.
5. Click Next.
6. Select Automatically assign using the user field selected earlier (Manager).
7. Click Save.

9. Click Go!

**Create the second step.**

1. Click New Approval Step under the Approval Steps related list.
2. For Name, enter Position Approver 1.
3. Click Next.
4. Select All records should enter this step.
5. Click Next.
6. Select Automatically assign to approver(s).
7. Select Related User from the picklist that appears, then select Approver 1 from the secondary picklist.
8. For When multiple approvers are selected, select Approve or reject based on the FIRST response.
9. For What should happen if the approver rejects this request? select Perform all rejection actions for this step AND all final rejection actions (Final Rejection).
10. Click Save.
11. Select No, I'll do this later. Take me to the approval process detail page to review what I've just created.
12. Click Go!

**Create the third approval step.**

1. Click New Approval Step under the Approval Steps.
2. For Name, enter Position Approver 2.
3. Click Next.
4. Select All records should enter this step.
5. Click Next.
6. Select Automatically assign to approver(s).
7. Select Related User from the picklist that appears, then select Approver 2 from the secondary picklist.
8. Click Save.
9. Select No, I'll do this later. Take me to the approval process detail page to review what I've just created.
10. Click Go!

**Create the Final Actions**

Now create the final actions: approval or rejection, along with their associated email alerts.

1. Under the Final Approval Actions related list, click Add New and select Field Update.
2. For Name, enter Approval Status to Approved.
3. For Field to Update, select Approval Status from the picklist.
4. Under Picklist Options, select A specific value, then choose Approved from the picklist.
5. Click Save & New.

Fill in the details of the second field update.

1. For Name, enter Status to Open.
2. For Field to Update, select Status from the picklist.
3. Under Picklist Options, select A specific value, then select Open from the picklist.
4. Click Save.

### Set up email alerts.

1. Under the Final Approval Actions related list, click Add New and select Email Alert.
2. For Description, enter Email position owner about approval.
3. For Email Template, select Position Approved. (Click Lookup iconto find it.)
4. For Recipient Type, select Owner.
5. In the Available Recipients column, select Position Owner then click Add arrowto move it to the Selected Recipients column.
6. Click Save.

## Create a Process for Submitting Positions for Approval

**Use Process Builder to Create a New Process Use Process Builder to set up the process.**

1. From Setup, enter Process Builder in the Quick Find box, then select Process Builder.
2. Click New.
3. Select Continue in Process Builder when prompted
4. For Process Name, enter Submit New Positions for Approval.
5. Under The process starts when, select A record changes from the picklist.
6. Click Save.
7. Click + Add Object and select Position from the Object picklist. (Do not worry if you see a warning message that Position already has active record-change processes.)
8. For Start the process, select when a record is created or edited.
9. Click Save.
10. Click + Add Criteria.
11. For Criteria Name enter Position Ready for Approval
12. For Criteria for Executing Actions, select Conditions are met.
13. Click + Add Row four times and set the conditions.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Operator** | **Type** | **Value** |
| Status | Equals | Picklist | New |
| Job Description | Does not equal | Global Constant | $GlobalConstant.N ull |
| Department | Does not equal | Global Constant | $GlobalConstant.N ull |
| Education | Does not equal | Global Constant | $GlobalConstant.N ull |
| Hiring\_Manager c | Does not equal | Global Constant | $GlobalConstant.N ull |

1. Click Save.
2. Under Immediate Actions,click Add Action and fill in the details.
3. Click Save,Activate,then Confirm.
4. Click Back To Setup

## Create a Candidate Rating Flow

### Create Radio Buttons for Experience

Begin creating the flow.

1. From Setup, enter Flows in the Quick Find box and select Flows.
2. Click New Flow.
3. Under Flow Types, select Screen Flow.
4. Click Create.
5. On the Canvas, click the Auto-Layout button, and select Freeform.
6. Add a screen element to prompt for the Review information by dragging Screen from the palette onto the flow window.
7. For Label, enter New Review.

Now add radio button choices to rate a candidate on a scale of 1 (poor) to 5 (excellent) for Experience, along with a text box for comments.

1. Click Radio Buttons from the Input section of the palette on the left.
2. Enter Experience in the Label field. (The API Name field will default.)
3. Set the Data Type to Number.
4. Click inside the Choice field, select +New Choice Resource, select Choice as the Resource Type, and fill in the details.

|  |  |
| --- | --- |
| **Field** | **Value** |
| API Name | Excellent |
| Choice Label | Excellent |
| Data Type | **Number** |
| Choice Value | 5 |

1. Click Done.
2. Click +Add Choice four times.
3. Click inside the second Choice field and select +New Choice Resource.
4. Select Choice as the Resource Type, and fill in the details.

|  |  |
| --- | --- |
| **Field** | **Value** |
| API Name | VeryGood |
| Choice Label | VeryGood |
| Data Type | **Number** |
| Choice Value | 4 |

1. Click Done.
2. Click inside the third Choice field and select +New Choice Resource.
3. Select Choice as the Resource Type, and fill in the details.
4. Click Done.
5. Click inside the fourth Choice field and select +New Choice Resource.
6. Select Choice as the Resource Type, and fill in the details.

|  |  |
| --- | --- |
| **Field** | **Value** |
| API Name | BelowAverage |
| Choice Label | BelowAverage |
| Data Type | **Number** |
| Choice Value | 2 |

1. Click Done.
2. Click inside the fifth Choice field and select +New Choice Resource.
3. Select Choice as the Resource Type, and fill in the details.
4. Click Done.
5. Click Text from the Input section of the palette on the left.
6. For Label, enter Experience Comments. (The API Name field will default.) We’ve got your first set of radio buttons, but don’t click Done on the screen just yet.

**Set a Start Element**

Now that you’ve created the flow, set its start element.

1. Connect the Start element to the New Review Screen element by clicking and holding the Start circle and then dragging your cursor to the New Review element.
2. You will see that the two elements are connected by a solid line.

Next add a Create Records element to the flow.

1. Drag the Create Records element from the palette onto the flow window.
2. For Label enter createReview, and let the API name default.
3. For How Many Records to Create, select one.
4. For How to Set the Record Fields select Use separate resources, and literal values.
5. For Object select Review.
6. Under Set Field Values for the Review click the Field box, and select Job\_Application\_ \_c.
7. For Value, click in the box, select +New Resource, and fill in the information.

|  |  |
| --- | --- |
| **Field** | **Value** |
| Resource Type | Variable |
| API Name | varJobAppId |
| Data Type | Text |
| Available for Input | Select |
| Available for Output | Select |

1. Click Done.
2. Confirm that the Value that aligns with the Job\_Application\_ \_c field now contains {!varJobAppId}.
3. Click +Add Field 8 times, then map the rest of the values according to the chart below.
4. Click Done.
5. Connect the New Review screen element to the createReview element by clicking and holding the New Review circle and then dragging your cursor to the createReview element.

**MODULE-5**

**[Build a Discount Approval Process]**

1. **Discount Approval Process - Agenda**
   1. [Prepare Your Org](https://trailhead.salesforce.com/content/learn/projects/build-a-discount-approval-process/prepare-your-org)
   2. [Create an Approval Process](https://trailhead.salesforce.com/content/learn/projects/build-a-discount-approval-process/create-an-approval-process)
   3. [Create Initial Submission Actions](https://trailhead.salesforce.com/content/learn/projects/build-a-discount-approval-process/create-initial-submission-actions)
   4. [Specify Final Approval and Rejection Actions](https://trailhead.salesforce.com/content/learn/projects/build-a-discount-approval-process/specify-final-approval-and-rejection-actions)
   5. [**Prepare Your Org**](https://trailhead.salesforce.com/content/learn/projects/build-a-discount-approval-process/prepare-your-org)

**Add a New User**

**Launch Your Trailhead Playground**

You'll be completing this hands-on project in your own personal Salesforce environment, called a Trailhead Playground. Get your Trailhead Playground now by first logging in to Trailhead, and then clicking **Launch** at the bottom of this page. Your playground opens in a new browser tab or window. Keep the playground window open while you do this project. After you complete the project steps in your playground, come back to this window and click **Verify step** at the bottom of this page.

Prior to setting up process automation, first set up Allison Wheeler with a user record so you can assign tasks to her later.

From Setup, enter

* + 1. **Users** in the Quick Find box, then select **Users**.
    2. Click New User. Complete the new user record with these details
    3. Uncheck **Generate new password and notify user immediately**.
    4. Click **Save**. **Create a Role**

One of the first projects Allison takes on is some restructuring of AW Computing’s sales department. She’s adding a sales manager who will serve as the official discount approver. Add that role to the hierarchy, reporting to her.

1. From Setup, enter **Roles** in the Quick Find box, then click **Roles**.
2. Click **Set Up Roles**.
3. Click **Expand All**.
4. Below VP, North American Sales, click Add Role, and then complete the new role details
5. Click **Save**.

Allison has hired Courtney Brown to fill the manager role. Add Courtney as a user, with the newly created role of manager, so you can assign tasks to her later.

1. From Setup, enter **Users** in the Quick Find box, then select **Users**.
2. Click New User. Complete the new user record with these details.
3. Uncheck Generate new password and notify user immediately.
4. Click **Save**.

Next, set up a Discount Rejected email template.

Navigate back to the initia**l Classic Email Templates** page, then click **New Template**.

Select the **Text** radio button, then click **Next**. Fill in these details:

Click **Save**.

Now that you’ve got the prep work done, you’re ready to create an approval process.

* 1. [**Create an Approval Process**](https://trailhead.salesforce.com/content/learn/projects/build-a-discount-approval-process/create-an-approval-process)

You’ve added key users, created custom fields, and made the necessary email templates, so your org is prepared. Now you can start creating a new approval process for your sales team. And that starts by using the setup wizard to set some criteria and specify the approvers who are responsible for responding to approval requests.

Here’s how to start the approval process setup.

From Setup, enter **Approval Processes i**n the Quick Find box, then select **Approval Processes**.

From the Manage Approval Process For picklist, select **Opportunity**.

From the Create New Approval Process picklist, select **Use Standard Setup Wizard**and fill in these new approval process details:

|  |  |
| --- | --- |
| **Field** | **Value** |
| Process Name | Discount Approval Process |
| Unique Name | [this field auto-populates] |
| Description | Automates opportunity discount approvals |

Click **Next**.

In the Use this approval process if the following picklist, select **criteria are met**, then fill in the criteria:

|  |  |  |
| --- | --- | --- |
| **Field** | **Operator** | **Value** |
| Opportunity: Discount Percentage | greater than | 0 |

Click **Next**, then complete the approver field and record editability properties with these values:

|  |  |
| --- | --- |
| **Field** | **Value** |
| Next Automated Approver Determined By | Manager |
| Use Approver Field of Opportunity Owner | Select |
| Record Editability Properties | Administrators ONLY can edit records during the approval process |

Click **Next**.

In Step 4, leave the Approval Assignment Email Template field blank.

On the What Would You Like To Do Now? page, select **I'll do this later, take me to the approval detail page to review what I've just created**.

Click **Go!**

Great work! Let’s move on to the next step, where you specify initial submission actions in the approval process.

* 1. [**Create Initial Submission Actions**](https://trailhead.salesforce.com/content/learn/projects/build-a-discount-approval-process/create-initial-submission-actions)

You’ve added key users, created custom fields, and made the necessary email templates, so your org is prepared. Now you can start creating a new approval process for your sales team. And that starts by using the setup wizard to set some criteria and specify the approvers who are responsible for responding to approval requests.

Here’s how to start the approval process setup.

From Setup, enter **Approval Processes i**n the Quick Find box, then select **Approval Processes**.

From the Manage Approval Process For picklist, select **Opportunity**.

From the Create New Approval Process picklist, select **Use Standard Setup Wizard**and fill in these new approval process details:

|  |  |
| --- | --- |
| **Field** | **Value** |
| Process Name | Discount Approval Process |
| Unique Name | [this field auto-populates] |
| Description | Automates opportunity discount approvals |

Click **Next**.

In the Use this approval process if the following picklist, select **criteria are met**, then fill in the criteria:

|  |  |  |
| --- | --- | --- |
| **Field** | **Operator** | **Value** |
| Opportunity: Discount Percentage | greater than | 0 |

Click **Next**, then complete the approver field and record editability properties with these values:

|  |  |
| --- | --- |
| **Field** | **Value** |
| Next Automated Approver Determined By | Manager |
| Use Approver Field of Opportunity Owner | Select |
| Record Editability Properties | Administrators ONLY can edit records during the approval process |

Click **Next**.

In Step 4, leave the Approval Assignment Email Template field blank.

Click **Next**, then determine the fields that will be displayed on the approval request page:

|  |  |
| --- | --- |
| **Field** | **Value** |
| Selected Field | Opportunity Name Opportunity Owner Amount  Discount Percentage |
| Display approval history information in addition to the fields selected above | Select |
| Security Settings | Allow approvers to access the approval page  only from within the salesforce.com application |

|  |  |
| --- | --- |
|  | (Recommended) |
| **Field** | **Value** |
| Submitter Type | Owner |
| Allowed Submitters | Opportunity Owner |
| Allow submitters to recall approval requests | Deselect |

Click **Next**, then specify the initial submitter by completing the submitter details in Step

Click **Save**.

On the What Would You Like To Do Now? page, select I'll do this later, take me to the approval detail page to review what I've just created.

Click Go!

Great work! Let’s move on to the next step, where you specify initial submission actions in the approval process.

* 1. **Create a Process for Submitting Positions for Approval**

Now that you’ve set up an approval process, continue to automate recruiting tasks by creating a process that automatically submits new positions for approval.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Operator** | **Type** | **Value** |
| Status | Equals | Picklist | New |
| Job Description | Does not equal | Global Constant | $GlobalConstant.Null |
| Department | Does not equal | Global Constant | $GlobalConstant.Null |

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Operator** | **Type** | **Value** |
| Education | Does not equal | Global Constant | $GlobalConstant.Null |
| Hiring\_Manager c | Does not equal | Global Constant | $GlobalConstant.Null |

Click Save.

|  |  |
| --- | --- |
| **Field** | **Value** |
| Action Type | Submit for Approval |
| Action Name | Submit Position for Approval |
| Object | Position c |
| Approval Process | Specific approval process: New Position Approval – New\_Position\_Approval |

Under Immediate Actions, click **Add Action** and fill in the details. Click **Save**, **Activate**, then **Confirm**.

Click .

You’re almost done setting automating processes for the recruiting team. Move on to the last step, where you create a flow with radio buttons that interviewers can use to easily rate candidates.

* 1. **Create a Candidate Rating Flow**

Right now, the AW Computing hiring managers have to manually input a number when they rate candidates. Ling Wu wants this process to be more user friendly, with radio buttons.

To set this up for her, create a flow. Flows, which you build using Flow Builder, automate business processes by executing logic, interacting with the Salesforce database, and collecting data from users.

**Create Radio Buttons for Experience**

Begin creating the flow.

* From Setup, enter Flows in the Quick Find box and select **Flows**.
* Click **New Flow**.
* Under Flow Types, select **Screen Flow.**
* Click **Create**.
* On the Canvas, click the Auto-Layout button, and select **Freeform**.

Add a screen element to prompt for the Review information by dragging **Screen** from the palette onto the flow window.

For Label, enter New Review.

Now add radio button choices to rate a candidate on a scale of 1 (poor) to 5 (excellent) for Experience, along with a text box for comments.

* Click **Radio Buttons** from the Input section of the palette on the left.
* Enter Experience in the Label field. (The API Name field will default.)
* Set the Data Type to **Number**.
* Click inside the Choice field, select **+New Choice Resource**, select **Choice** as the Resource Type, and fill in the details.

|  |  |
| --- | --- |
| **Field** | **Value** |
| API Name | Excellent |
| Choice Label | Excellent |
| Data Type | **Number** |
| Choice Value | 5 |

* Click **Done**.
* Click **+Add Choice** four times.
* Click inside the second Choice field and select **+New Choice Resource**.

Select **Choice** as the Resource Type, and fill in the details.Click **Text** from the Input section of the palette on the left.

For Label, enter Experience Comments. (The API Name field will default.)

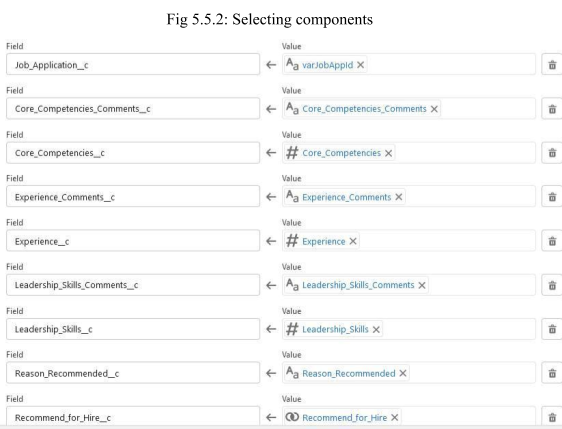
You’ve got your first set of radio buttons, but don’t click Done on the screen just yet.

Create Radio Buttons for Leadership Skills

Now that radio buttons and text screen elements for Experience are added to the



flow, repeat the process for Leadership Skills. Verify that your screen looks something like this.



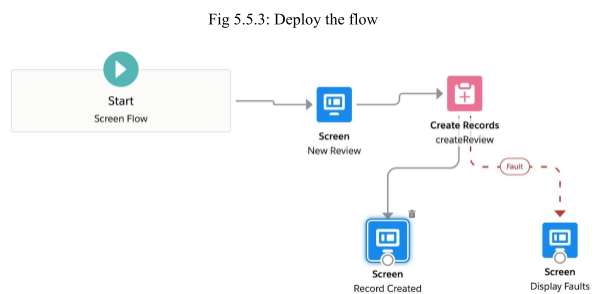
* Click **Done**.
* Connect the New Review screen element to the createReview element by clicking and holding the New Review circle and then dragging your cursor to the createReview element.

**Add Screen Elements**

Next, add a screen element to display a success message.

* Drag **Screen** from the palette onto the flow window.
* For Label, enter Record Created. (The API Name field will default.)
* Under Display on the left, click **Display Text**.
* For API Name, enter RecordCreated.
* In the text box enter Your review record has been created successfully. Leave the Insert a resource field above the text box empty.
* Click **Done**.

Connect the createReview element to the Record Created element. Connect the createReview element to the Display Faults element.Verify that your Flow looks

something like this.

### Deploy the Flow

Now that the New Review Flow is set up, create a custom button for launching it.

* From Setup, click **Object Manager** and select **Job Application**.
* Click **Buttons, Links, and Actions**.
* Select **New Button or Link**.
* For Label, enter New Review.
* Select **Detail Page Button** as the display type.
* Click **Save**, then click **OK** on the popup. Now add the New Review button to the page layout.

While still viewing Job Application in Object Manager, click **Page Layouts**.

* Click  next to Job Application Layout and select **Edit**.
* Click **Buttons** in the page layout editor.
* Drag the **New Review** button to the Custom Buttons section of the Job Application
* Click **Save**.

By setting up processes, creating an approval process, and building a flow, you’ve automated a number of tasks for the recruiting team.

**MODULE-6**

# [Quick Start: Lightning App Builder]

## Quick Start: Lightning App Builder - Agenda

* 1. Create Your First Page
  2. Add M+ore Components
  3. Add Quick Actions and Activate the App
  4. **Create Your First Page**

### Introduction

Lightning App Builder lets developers and business users build beautiful custom user interfaces that are designed to work perfectly on your desktop and mobile devices, all without writing a single line of code. In this Quick Start, you'll build a geolocation app for sales reps in the field using the Lightning App Builder. You'll use a pre-built Lightning Component and the drag-and-drop interface of Lightning App Builder to create this app.

### Create Your First Page

A Lightning Page is a container for Lightning Components. Create your first page and add a component to it.

* + 1. If you haven’t already, log in to Trailhead, then launch your Trailhead Playground by clicking Launch at the bottom of this page. This opens your Trailhead Playground in a new tab.
    2. If you're not already on the Setup home page, click settings and select Setup to launch Setup in a new tab.
    3. Enter Lightning App Builder in Quick Find and select Lightning App Builder.
    4. Under Lightning Pages, click New.
    5. Select App Page then click Next.
    6. In the Label field, type Field Sales App and click Next.
    7. In Create a New Lightning Page, select Two Regions.
    8. Click Finish.

Now that you have a page, you can add components. For your first component, drag a Filter List component onto the page.

* + 1. In the Desktop drop-down list, click Tablet - Portrait. The canvas's preview layout changes from the standard single column to two columns.
    2. From the Standard Components menu on the left, drag the ListView to the left column.
    3. Set the properties of this component using the Properties list in the right sidebar.
    4. In the Object drop-down list, select Account. 13.In the Filter drop-down list, select My Accounts.

1. In the Number of Records to Display field, enter 7.
2. 4. Click Save and then click Not Yet in the popup window. We will activate this page in a later step.

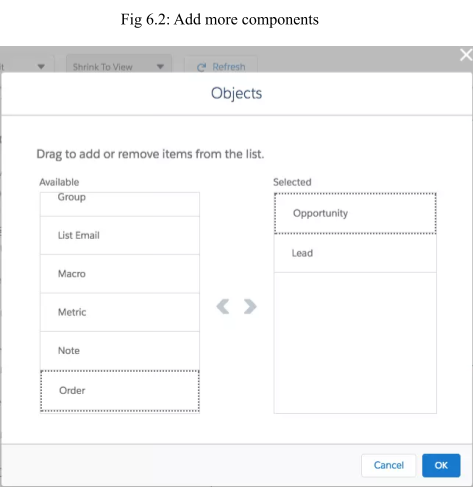
## Add More Components

### Add Components to the Page

You can now add more components to your page.

This time you'll add a Recent Items component, which you configure to display recent opportunities and leads.

1. From the Standard Components menu on the left, drag the Recent Items component to the right column.
2. In the property inspector on the right, type Opportunities & Leads for the label.
3. Click Select under Objects, and highlight API Anomaly Event Store from the Selected column on the right to the Available column on the left. Click the left arrow to remove
4. Click Opportunity from Available and click the right arrow to add to Selected. Repeat this for the Lead object.
5. Click OK 21.Click Save



## Add Quick Actions and Activate the App

### Add Quick Actions and Configure the Page

Quick actions allow users to quickly create and modify records. After you add the actions, you activate the Lightning Page as an app, which makes it available to mobile users.

22.In the right sidebar, click Page to configure the app properties. 23.At the bottom, click Select under Actions.

1. Click Log a Call and use the right arrow to add the quick action to the Selected list. Repeat for New Case, New Lead, and New Task.
2. Click OK to add the actions to your Lightning Page, and then click Save.

The Actions property at the bottom of the inspector now shows the actions that you added.

Activate the app, which automatically creates tabs and adds them to the Salesforce mobile app navigation.

1. Click Activation.
2. On the Page Settings tab, type Field Sales App for the App Name. It should already be defaulted in for you. Leave all other defaults as is.
3. On the Mobile Navigation tab, click the Add page to app button and then drag the Field Sales App icon and place it below the Today icon.
4. Click Save to activate.

### Check Out Your New App

Congratulations, you just built a basic single-page Lightning App. Let's check it out and see how it looks.

1. Click Back in the upper left to exit the Lightning App Builder.
2. Click Apps and type Field Sales into the text box. Click Field Sales App.

Behold your new Field Sales App!

**MODULE-7**

# [Quick Start: Process Builder]

## Quick Start: Process Builder - Agenda

* 1. Create a New Process on the Account Object
  2. Add Process Criteria
  3. Add Your Process Action
  4. Test Your Process
  5. **Create a New Process on the Account Object**

### Introduction

Process Builder is a workflow tool that helps automate business processes without writing a single line of code. For example, imagine that a company you do business with (which we call an Account in Salesforce), changes its location. You'd want a way to automatically update the business address of all the people that work at that company (your Contacts).

In this Quick Start, you'll do exactly that: create a new process that updates Contact records whenever the Account billing address changes. As you'll see, it's quite easy and doesn't require a shred of code.

### Create a New Process on the Account Object

You first create a process and then select the object on which the process runs. You also make sure the process kicks off whenever a record is edited because you’re going to change the business address in a moment.

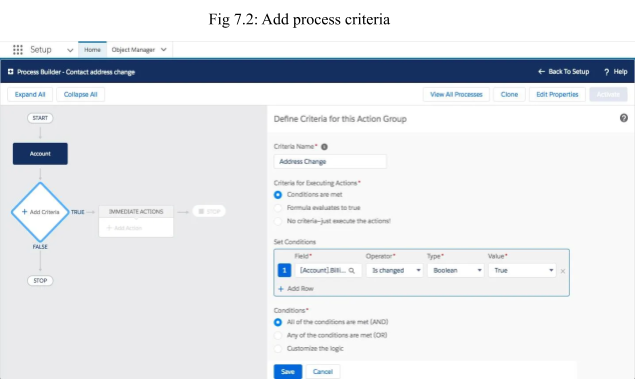
* + 1. Click settings and select Setup. This launches Setup in a new tab.
    2. From Setup, enter Builder in the Quick Find box, and select Process Builder.
    3. Click New.
    4. Click the Continue with Process Builder button.
    5. For Process Name, type Contact address change.
    6. For The process starts when, select A record changes, and click Save.
    7. Click + Add Object.
    8. In the right window, select Account from the Object drop-down list.
    9. For Start, the process selects when a record is created or edited. 10.Click Save

## Add Process Criteria

### Create Criteria

You now define the criteria that determine when this process runs.

* + 1. Click Add Criteria.
    2. For Criteria Name, type Address Change.
    3. For Criteria for Executing Actions, keep it set to Conditions are met.
    4. For Set Filter Conditions, click Find a field..., select Billing Street and click Choose.
    5. Set Operator to Is Changed and set Value to True.
    6. For Conditions, keep it set to All of the conditions are met (AND).
    7. Click Save.



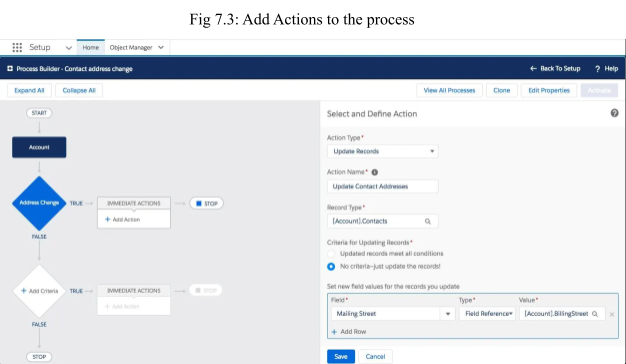
In plain language, this all means "run this process when the Billing Street changes."

## Add Your Process Action

### Create an Action

At this point, you've defined a process and told it when to fire, but you haven't told it what to do. In this step, you define what happens when Billing Street changes.

* + 1. Under the Immediate Actions box, click + Add Action.
    2. In the Action Type drop-down list, select Update Records.
    3. For Action Name, type Update Contact Addresses.
    4. For Record Type, click the radio button next to Select a record related to the Account, then scroll down and select Contacts, and click Choose. This is pretty powerful stuff, so pause and take note. Process Builder allows you to choose not just fields on Accounts, but fields that are related to Accounts.
    5. For Criteria for Updating Records, keep it set at No criteria—just update the records!
    6. Under Set new field values for the records you update, click Find a field..., and then scroll down and select Mailing Street.
    7. Select Field Reference for the Type.
    8. For Value, select Billing Street as the Account field and click Choose.
    9. Click Save.
    10. Click Activate and then click Confirm.



Some important things to note before you move on:

* This task adds only one action, but you could add multiple actions for one criterion.
* Did you notice that Accounts have a field called Billing Street, but that Contacts have a field called Mailing Street? That's OK, it's a different field name, but you'll use the same value.
* Once a process is activated, it cannot be edited. If you need to make edits to an activated process, follow these steps: (1) Clone the process as a version of the current process; (2) make changes to the cloned process;

(3) activate the new version.

## Test Your Process

### Check the Process

If everything works correctly, whenever an account has a change of address, the address change is made for everyone who works for that company. Let’s test that.

* + 1. Open the App Launcher. Select Contacts.
    2. Click Recently Viewed and select the All Contacts list, then click Tim Barr.
    3. On Tim Barr’s contact record, click Details.
    4. On Tim Barr’s detail page, note the mailing address [1] and account name [2].
    5. Click the account name, Grand Hotels & Resorts Ltd. This opens the account for Grand Hotels & Resorts Ltd.
    6. On the Grand Hotels & Resorts Ltd. account record, click Details.
    7. Click Edit.
    8. Change the billing street address and click Save.
    9. Click Related.
    10. Scroll down to Contacts and click Tim Barr. Click Details. On his detail page, notice that his mailing address has been updated

**MODULE-8**

# [Data Management]

## Data Management - Agenda

* 1. Import Data
  2. Export Data
  3. **Import Data**

Learning Objectives

After completing this unit, you'll be able to:

* Describe and compare the different options for importing data into Salesforce.
* List the steps involved in preparing and importing data from a sample .csv file using the Data Import Wizard.

### Introduction to Data Import

You can easily import external data into Salesforce. Supported data sources include any program that can save data in the comma delimited text format (.csv).

Salesforce offers two main methods for importing data.

* + **Data Import Wizard**—this tool, accessible through the Setup menu, lets you import data in common standard objects, such as contacts, leads, accounts, as well as data in custom objects. It can import up to 50,000 records at a time. It provides a simple interface to specify the configuration parameters, data sources, and the field mappings that map the field names in your import file with the field names in Salesforce.
  + **Data Loader**—this is a client application that can import up to five million records at a time, of any data type, either from files or a database connection. It can be operated either through the user interface or the command line. In the latter case, you need to specify data sources, field mappings, and other parameters via configuration files. This makes it possible to automate the import process, using API calls.

**Use the Data Import Wizard When:**

* + You need to load less than 50,000 records.
  + The objects you need to import are supported by the wizard.
  + You don’t need the import process to be automated.

**Use Data Loader When:**

* + You need to load 50,000 to five million records. If you need to load more than 5 million records, we recommend you work with a Salesforce partner or visit the AppExchange for a suitable partner product.
  + You need to load into an object that is not supported by the Data Import Wizard.
  + You want to schedule regular data loads, such as nightly imports.

Data Loader uses the SOAP API to process records. For faster processing, you can configure it to use the Bulk API instead. The Bulk API is optimized to load a large number of records simultaneously. It is faster than the SOAP API due to parallel processing and fewer network round-trips.

### Prepare for Data Import

Follow these steps before you start importing any data.

* + 1. Use your existing software to create an export file. You'll use this exported data file to now *import* the data into Salesforce.
    2. Clean up the import file for accuracy and consistency. This involves updating the data to remove duplicates, delete unnecessary information, correct spelling and other errors, and enforce naming conventions.
    3. Compare your data fields with the Salesforce fields you can import into, and verify that your data will be mapped into the appropriate Salesforce fields. You might need to fine-tune the mapping before starting the import. For details, see Field Mapping for Data Sources in the online help.
    4. Make any configuration changes required in Salesforce to handle the imported data. For example, you might need to create new custom fields, add new values to picklists, or temporarily deactivate workflow rules.

View the second video "Data Import: Clean Up Your Import File" in the How To Import Data into Salesforce Series for more information on cleaning up your import file.

### Use the Data Import Wizard

Once you have created an export file and cleaned up the data for import, follow these steps to import data using the Data Import Wizard.

1. Start the wizard.
   1. From Setup, enter Data Import Wizard in the Quick Find box, then select **Data Import Wizard**.
   2. Review the information provided on the welcome page, then click **Launch Wizard!**
2. Choose the data that you want to import.
   1. To import accounts, contacts, leads, solutions, person accounts, or campaign members, click **Standard Objects**. To import custom objects, click **Custom Objects**.
   2. Specify whether you want to add new records to Salesforce, update existing records, or add and update records simultaneously.
   3. Specify matching and other criteria as necessary. Hover over the question marks for more information about each option.
   4. Specify the file that contains your data. You can specify your data file by dragging the CSV to the upload area of the page or by clicking the CSV category you’re using and then navigating to and selecting the file.
   5. Choose a character encoding method for your file. Most users can accept the default character encoding.
   6. Click **Next**.
3. Map your data fields to Salesforce data fields. The Data Import Wizard tries to map as many of your data fields as possible to standard Salesforce data fields. If Salesforce can’t automatically map fields, however, you do it manually. Unmapped fields are not imported into Salesforce. To see a list of standard Salesforce data fields, from Setup, at the top of the page, click **Object Manager**. Click the object whose fields you’re interested in, and click **Fields & Relationships**. For example, if you want to see a list of standard Salesforce fields for leads, click **Object Manager** | **Lead** | **Fields & Relationships**.
   1. Scan the list of mapped data fields and locate any unmapped fields.
   2. Click **Map** to the left of each unmapped field.
   3. In the Map Your Field dialog box, choose the Salesforce fields you want to map to and click **Map**. The Map Your Field dialog box also gives you the option of saving data from unmapped fields in a general notes field for accounts and contacts. To do so, choose Account Note or Contact Note from the Map To drop-down list and click **Map**.
   4. To change mappings that Salesforce performed automatically, click **Change** to the left of the appropriate field, then choose the Salesforce fields you want to map to and click **Map**.
   5. Click **Next**.
4. Review and start your import.
   1. Review your import information on the Review page. If you still have unmapped fields that you want to import, click **Previous** to return to the previous page and specify your mappings.
   2. Click **Start Import**.
5. Check import status. From Setup, enter “Bulk Data Load Jobs” in the Quick Find box, then select **Bulk Data Load Jobs**. The user who starts the data import receives a status email when the import is completed.

This information can help you integrate your imported data into Salesforce.

* + New Values for Picklists and Multi-Select Picklists—If you import a picklist value that doesn’t match an existing picklist value:
    - For an unrestricted picklist, the Data Import Wizard uses the value that’s in the import file.
    - For a restricted picklist, the Data Import Wizard uses the picklist’s default value.

## Export Data

Learning Objectives

After completing this unit, you'll be able to:

* Describe and compare the two methods of exporting data from Salesforce.
* Export data manually using the Data Export Service.
* Set up automatic export of data on a weekly or monthly schedule.

### Introduction to Data Export

You can easily export data from Salesforce, either manually or on an automatic schedule. The data is exported as a set of comma-separated values (CSV) files. Data export tools provide a convenient way to obtain a copy of your Salesforce data, either for backup or for importing into a different system.

Salesforce offers two main methods for exporting data.

* **Data Export Service**—an in-browser service, accessible through the Setup menu. It allows you to export data manually once every 7 days (for weekly export) or 29 days (for monthly export). You can also export data automatically at weekly or monthly intervals. Weekly exports are available in Enterprise, Performance, and Unlimited Editions. In Professional Edition and Developer Edition, you can generate backup files only every 29 days, or automatically at monthly intervals only.
* **Data Loader**—a client application that you must install separately. It can be operated either through the user interface or the command line. The

latter option is useful if you want to automate the export process, or use APIs to integrate with another system.

### Using the Data Export Service

Follow these steps to export data using the Data Export Service.

* + 1. From Setup, enter Data Export in the Quick Find box, then select **Data Export** and **Export Now** or **Schedule Export**.

o The **Export Now** option prepares your files for export immediately. This option is only available if enough time has passed since your last export.

o The **Schedule Export** option allows you to schedule the export process for weekly or monthly intervals.

* + 1. Select the desired encoding for your export file.
    2. If you want images, documents, attachments, and so on included in your data, select the appropriate options.
    3. Select Replace carriage returns with spaces to have spaces instead of carriage returns or line breaks in your export files. This is useful if you plan to use your export files for importing or other integrations.
    4. If you're scheduling your export, select the frequency (only available for organizations with monthly exports), start and end dates, and time of day for your scheduled export.
    5. Under Exported Data, select the types of data to include in your export. We recommend that you select **Include all data** if you’re not familiar with the terminology used for some of the types of data.
    6. Click **Start Export** or **Save**. Salesforce creates a zip archive of CSV files and emails you when it's ready. Exports will complete as soon as possible, however, we can't guarantee the date and time the export will complete. Large exports are broken up into multiple files. Follow the link in the email or click **Data Export** to download the zip file. Zip files are deleted 48 hours after the email is sent.

**MODULE-9**

# [Event Monitoring]

## Event Monitoring - Agenda

* 1. Getting started with event monitoring
  2. Query event log files
  3. Download and visualize event log files.
  4. **Getting Started with Event Monitoring**

Learning Objectives

After completing this unit, you’ll be able to:

* + - Describe the event types supported by Event Monitoring.
    - Define event log files.
    - State at least three use cases for Event Monitoring.
    - Describe the application programming interface (API)-first approach to development.

### What is Event Monitoring ?

Everyone knows that being a detective is one of the coolest jobs you can have. Well, hold on to your magnifying glass because your job as a Salesforce admin is about to get a whole lot cooler. With Event Monitoring, you can be the investigator your organization always needed.

Event Monitoring is one of many tools that Salesforce provides to help keep your data secure. It lets you see the granular details of user activity in your organization. We refer to these user activities as events. You can view information about individual events or track trends in events to swiftly identify abnormal behavior and safeguard your company’s data.

So, what are some of the events that you can track? Event Monitoring provides tracking for many types of events, including:

* + - Logins
    - Logouts
    - URI (web clicks in Salesforce Classic)
    - Lightning (web clicks, performance, and errors in Lightning Experience and the Salesforce mobile app)
    - Visualforce page loads
    - Application programming interface (API) calls
    - Apex executions
    - Report exports

There are 50 event types that can be consumed; check out the Report Event Type document in this unit’s Resources section for the full list.

All of these events are stored in event log files. An event log file is generated when an event occurs in your organization and is available to view and download after 24 hours. The event types you can access and how long the files remain available depends on your Salesforce edition.

* + - Developer Edition organizations have free access to all log types with 1-day data retention.

Enterprise, Unlimited, and Performance Edition organizations have free access to the insecure external assets, login, logout, and total API usage event log files with 1-day data retention. For an extra cost, you can access all log file types with 30-day data retention.

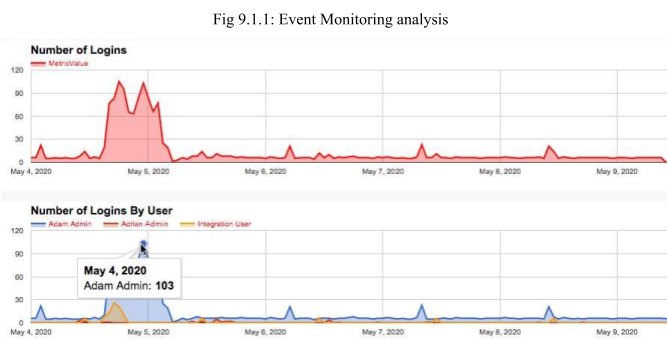
So, how can you use event log files to become an all-knowing Salesforce super-sleuth? Let’s take login activity as an example. We talk about accessing, downloading, and visualizing event log files later on. For now, assume that we did these steps and produced this graph of login activity.

You can see that an unusually high number of logins to the organization occurred between May 4 and May 5. But how do you figure out exactly what happened during that time period? Luckily, Event Monitoring provides several ways for you to dig into this data. In this case, you might want to break down the number of logins by user.

Adam Admin logged in 103 times! Something is definitely suspicious. You can continue to break this data down to see things like how many distinct IP

addresses a user logged in from. This information helps you pinpoint whether an outside party compromised a user’s account or whether a user is up to no good.

You’re probably beginning to see the power of Event Monitoring, but let’s consider some other uses.



* + - Monitor data loss. Imagine that a sales rep leaves your company and joins a major competitor. Later, you find out that your organization is losing deal after deal to this other company. You suspect that your former employee downloaded a report containing leads and shared it with the competition. If you’d been using Event Monitoring, you could have caught this bad behavior before it cost your company sales.
    - Increase adoption. Event Monitoring isn’t just for catching your users’ bad behavior. It can also alert you to parts of your organization that aren’t performing well. For example, you just rolled out a new Visualforce page in your organization that combines accounts and contacts and allows end users to add custom fields. Without any metrics, it’s difficult to tell how users are interacting with this page—if at all. Event Monitoring helps you figure out which parts of your organization need increased adoption efforts and identify areas that need redevelopment.
    - Optimize performance. Sometimes, it’s hard to determine the cause of slow page performance in your organization. Imagine that your company has an office in San Francisco and one in London. The users in London tell you that their reports are running slowly or even timing out. You can use Event Monitoring to determine whether the cause is related to a network issue in London or with the way your app is configured.

These cases are just a few ways that you can use Event Monitoring to keep your organization secure and running smoothly. Check out all the event types to discover what else you can do.

### A Quick Note About the API

If you’re an admin, working with the API can be daunting. We won’t go over all the nitty-gritty details in this module, but let’s take a minute to review some basics. API stands for application programming interface. You can think of it as a bridge between an application (in our case, Salesforce) and the database. Two important terms to remember when working with the API are:

* + - Objects: Almost every object in the user interface is also an object in the API (for example, Account or Case). The API also has several objects that you can’t use in the user interface.
    - Fields: The fields you’re used to seeing in the user interface are also fields in the API (for example, the Account Name field in the user interface becomes the Name field in the API).

Sometimes, the user interface doesn’t provide you with every possible access point to your data. That’s why the API is so important. Salesforce encourages what’s called an API-first approach to development. API-first means that, before you develop an application’s user experience, you want to pay attention to the underlying API. The API lets you use your data in ways that aren’t possible in the user interface. Considering the API in the initial planning stages lets you develop a more robust application.

Event Monitoring is an API-only feature. Each organization’s event log files are stored in an API standard object called EventLogFile. If all this information sounds a little confusing, don’t worry. We go through everything step-by-step in the next units

## Query Event Log Files

Learning Objectives

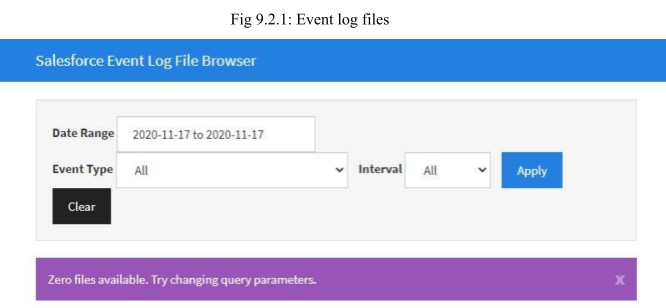
After completing this unit, you’ll be able to:

* + - Query an EventLogFile object using Developer Console.
    - View events in Salesforce Event Log File (ELF) Browser.
    - Learn about EventLogFile event types.

In this module, we assume you are a Salesforce admin with the proper permissions to enable event monitoring. If you’re not an admin for Salesforce, that’s OK. Read along to learn how your admin would take the steps in a production org. You can also follow along using a Trailhead Playground, but it takes 24 hours for events to appear in log files.

### View Events in Salesforce Event Log File Browser

The Salesforce Event Log File (ELF) Browser is a Salesforce-connected web app that allows quick access to event log files. With the ELF Browser, you can easily find and download events from various time periods without a line of code. The data in the files you get from the browser can even be visualized using Tableau CRM. See the [Event Monitoring Analytics App](https://trailhead.salesforce.com/content/learn/modules/event_monitoring_analytics) module for more information on that.



If no reports have been exported from your organization in the past 24 hours, the totalSize field has a value of zero. Remember that it takes 24 hours for events to become available. You can export a report from your organization and try again tomorrow.

1. Log in to your org.
2. Navigate to the ELF Browser application by clicking this link: [https://salesforce-elf.herokuapp.com](https://salesforce-elf.herokuapp.com/).
3. Click Production Login.
4. Set the start Date.
5. Select an event type for your search, or leave event type set to All.
6. Click Apply.

### Query Event Log Files in Developer Console

Let’s consider an example: A sales rep named Rob Burgle left your company a few weeks ago and joined a major competitor. All of a sudden, you start losing deals to this other company. You suspect that Rob downloaded a report containing confidential lead information and shared it with his new employer. Normally, you wouldn’t be able to confirm your suspicions. But with Event Monitoring, you can gather all the evidence you need to set the story straight. Let’s look at how this process works.

It takes only a couple of clicks to open the Developer Console from Lightning Experience. The Developer Console is an integrated development environment with a collection of tools you can use to create, debug, and test applications in your Salesforce org.

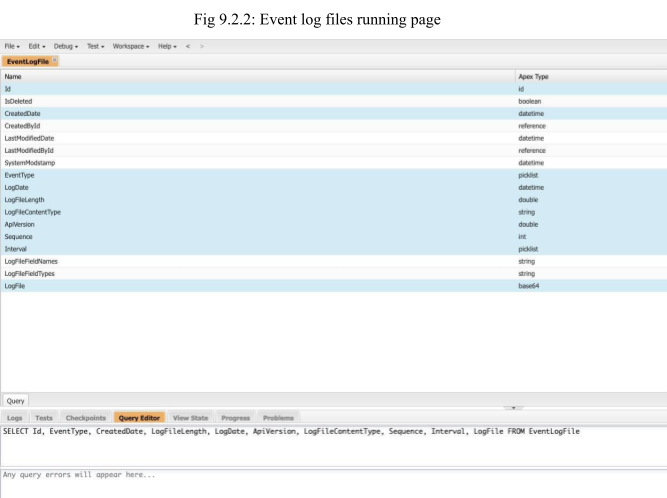
To open the Developer Console from Lightning Experience:

1. Click .
2. Click Developer Console.

Now we’re ready to open the EventLogFile to query the information.

1. Click File | Open.
2. Under Entity Types, select Objects.
3. In the Filter the repository field, type EventLogFile.
4. Select EventLogFile under Entities.
5. Click Open.

Next, select the fields for your query and click the Query button.



Finally, click the Execute button to complete the query.

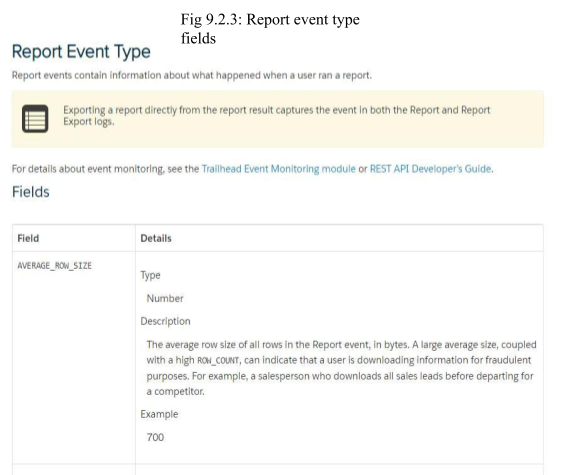
You can use the Query Editor in the Developer Console to execute a Salesforce Object Query Language (SOQL) query. The History pane displays your last 10 queries for quick reuse. Results are displayed in a Query Results grid.

The [Query Results grid](https://help.salesforce.com/articleView?id=code_dev_console_tab_query_editor_query_results_grid.htm&type=5) displays each record as a row. You can open, create, update, and delete records without leaving the Developer Console.

### Event Log File Documentation for Event Types

The EventType field in the EventLogFile object supports events. Every event type is documented in the Salesforce Object Reference. In it, you can see the fields and their descriptions, and sample queries to use.

For example, for the DB\_TOTAL\_TIME field, the event type Number is the time in nanoseconds for a database round trip, which includes time spent in the JDBC driver, network to the database, and DB\_CPU\_TIME. As an admin, you can compare this field to CPU\_TIME to determine whether performance issues are occurring in the database layer or in your own code.



Report events contain information about what happened when a user ran a report—things like the date and time, the report name, the records referenced in the report, the number of rows and columns, and the originating user, among many others.

You can also use the REST API to interact with Event Monitoring logs; Event Monitoring is accessed through the Lightning Platform SOAP API and REST API by way of the EventLogFile object. So, you’re able to integrate log data with your own back-end storage and data marts to correlate data from multiple orgs and across disparate systems.

In the next unit, let’s explore how to download and visualize event log files to get one step closer to providing insight into malicious user behavior using Event Monitoring.

## Download and Visualize Event Log Files

Learning Objectives

After completing this unit, you’ll be able to:

* + - Download an event log file.
    - Describe the structure of event log files.
    - Identify an application for downloading event log files without writing code using the Event Log File (ELF) Browser.
    - Explain how to use a cURL or Python script for downloading data.
    - Identify options for visualizing event log file data.

### Download Event Log Files Using Python

If you need a more programmatic way of downloading your organization’s event log files, you can use Python scripts. One advantage of using a Python script over a cURL script is that it’s easier for Windows users to work with, but it’s also suitable for Mac and Linux users.

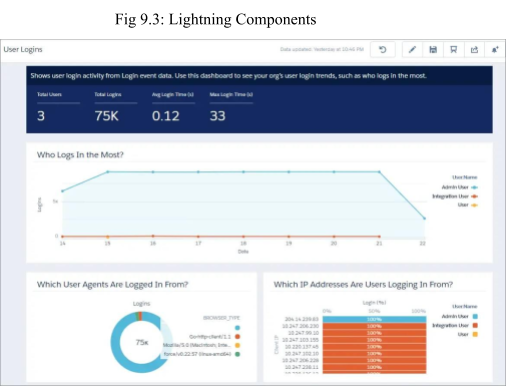
Python is easy to understand, even if you’re not an experienced programmer. Some setup is required, but after that you can easily run your download script. For more information and to download the code, see [this post](https://www.salesforcehacker.com/2015/03/elfpy-tasty-little-script-for.html).

### Visualize Event Log File Data

Now that you’ve taken the time to learn about event log files and how to download them from Salesforce, it’s time to talk about visualizing your data. Searching for a specific piece of information in thousands of rows in a spreadsheet is like searching for a needle in a haystack. Most of the time, it’s not useful to look for a single instance of a report export or user login. You’re probably more interested in noticing behavior that’s out of the ordinary. To get immediate insights into your organization’s inner workings, you can regularly download your event log files and create visual representations of your data.

**Event Monitoring Analytics app:** This Analytics app is a way to get insights into your Event Monitoring data without ever leaving the Salesforce Platform.

Your data is automatically loaded from Salesforce to the app, so you always get the most recent (and most stunning) visualization of what’s going on in your org. The app provides a collection of dashboards that use pre-integrated event data, so it’s a great way to get started with Event Monitoring.



**Splunk App for Salesforce:** This app lets you analyze and visualize your organization’s use of Salesforce and gain insights into security, performance, and user behavior. The Splunk Add-on for Salesforce lets a Splunk software admin collect different types of data from Salesforce using REST APIs. It also provides the inputs to use with other Splunk apps, such as Splunk Enterprise Security.

**FairWarning:** This user activity monitoring solution is purpose-built to translate and correlate Salesforce log files across Event Monitoring, real-time streams, reference objects, and Change Data Capture (CDC) events. In doing so, it allows them to provide user-centric insights and real-time alerts on abnormal behavior. And that helps you proactively identify threats and mitigate risk to your Salesforce dat

**CONCLUSION**

By completing this internship, a Recruiting App has been built with the help of Salesforce platform, by using Lightning app builder and flow builder concepts which helps in learning the importance of using automating systems to automate the business process in an application and its components which include creating a custom home pages, working with custom lightning pages, Flow resources and variables, adding logic and actions to a flow, adding process criteria and testing the process criteria are practiced thoroughly.

The concepts of process builder and app builder helps to understand the process of automating a business process. The Data management and event monitoring concepts provide a practical way of managing,importing, exporting and monitoring data on cloud. The application also used formulas and validation rules to implement certain mechanisms in the data model. These are helpful in manipulating data, such as storing data only when certain criteria mentioned in the validation rule is met. These validation rules ensure only the suitable and appropriate data is being stored in the data model created for the application.