Unit: Develop Without Code

* Describe the benefits of the metadata-driven development model.
* Define and give examples of the no-code and low-code development approaches.

The data you see in the UI is an abstraction of the database that is driven by the platform’s metadata-aware architecture. These objects are the database tables. You have **standard objects** that come with Salesforce by default and **custom objects** that you build yourself.

Metadata forms the structure of your org. Whether you’re defining fields, business processes, or something more complex, metadata holds your configuration.

This **metadata-driven development model** is one of the key differences between developing on the platform and developing outside of Salesforce.

The platform is metadata-aware and it can auto-generate:

* Dialogs, record lists, detail views, and forms
* Full CRUD functionality

Schema Builder provides both coders and non-coders with an easy way to visualize and configure an app’s data model

Process Builder is a low-code tool.

No Code

Salesforce offers a host of tools for point-and-click—or **declarative**—development.

Low Code

Some development tasks, like writing validation rules or hooking up components with UI elements, are considered low code.

The no-code and low-code development capabilities allows you to build more in less time. You can leave the declarative development tasks to non-coders.

**=> QUIZ**

**What's the relationship between objects, fields, and records and Salesforce's relational database?**

**Ans:** The data you see in the UI is an abstraction of the database that is driven by the platform’s metadata-aware architecture.

**How does the metadata-driven architecture support declarative development?**

**Ans:** The no-code and low-code development capabilities allows you to build more in less time. You can leave the declarative development tasks to non-coders.

Unit: Code with Salesforce Languages

* Identify the benefits of Lightning components.
* Describe how Visualforce is used in Lightning Experience.
* Outline the ways Apex is used to support Lightning components and Visualforce.

There are three core programmatic technologies to learn about as a Salesforce developer.

* Lightning Component Framework: A UI development framework similar to AngularJS or React.
* Apex: Salesforce’s proprietary programming language with Java-like syntax.
* Visualforce: A markup language that lets you create custom Salesforce pages with code that looks a lot like HTML, and optionally can use a powerful combination of Apex and JavaScript.

**Lighting Components**

* The Lightning Component framework is a user interface development framework for desktop and mobile. It’s a component-based approach to UI development.
* Lightning components use client-side JavaScript controllers and server-side Apex controllers.
* Lightning components are mobile-ready.

Spring 2019 - API version 45.0 Release

You can build Lightning components using two programming models: the Lightning Web Components model and the original Aura Components model. Lightning web components are custom HTML elements built using HTML and modern JavaScript. Lightning web components and Aura components can coexist and interoperate on a page.

Developer Console

The Developer Console is the Salesforce integrated development environment (IDE) that you can use to develop, debug, and test code in your org.

**Apex**

Apex is a Java-like syntax language developed by Salesforce.

Invoking Methods

In Apex the @InvocableMethod annotation will have a label to allow other tools like Process Builder to execute the method.

SOQL (Salesforce Object Query Language)

You can use SOQL to read records from the database in your code.

**Visualforce**

Visualforce lets you create and customize pages in Salesforce as well as integrate with other standard web technologies, including HTML, CSS, and JavaScript.

The difference between Visualforce and Lightning Components

With Lightning components, you’re developing components that can be pieced together to create pages. With Visualforce, you’re developing entire pages at once.

Visualforce markup

HTML markup, CSS, and JavaScript.

SLDS (Salesforce Lightning Design System)

Lets you style your pages so they match the look and feel of Salesforce’s new interface, Lightning Experience

*Visualforce pages can also use server-side Apex controllers.*

**=> QUIZ**

**What types of elements do you see in the XML markup for Aura components?**

**Ans:** Static HTML tags and Javascript

**What's one situation where it's better to use Lightning components instead of Visualforce?**

**Ans:** Your project will run primarily on mobile devices

**What's true about Apex controllers?**

**Ans:** Lighting components use server-side Apex controllers

Unit: Extend the Salesforce Platform

* List the Salesforce APIs.
* Explain how Heroku and Salesforce are related.
* Identify ways Salesforce interacts with IoT and bots.

**Heroku**

* Heroku is a web development platform that lets you quickly build, deploy, and scale web apps.
* With Heroku you have a lot of flexibility in how you write your app.
* Heroku is built on Amazon Web Services (AWS) - No concerns with the infrastructure of standard web app development
* *Heroku Connect* unifies your Salesforce data with your Heroku Postgres data so you don’t have to manage moving information across platforms. - No concerns with data storage

**=> QUIZ**

**For sending secure notifications, which API is your best bet?**

**Ans:** Streaming API

**Which Heroku service allows you to unify your Salesforce data with Postgres data?**

**Ans:** Heroku Connect