**DAILY ASSESSMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **14/07/2020** | **Name:** | **Dhavala** |
| **Course:** | **Coursera**  **Mathematics for Machine learning: Linear algebra** | **USN:** | **4AL17EC027** |
| **Topic:** | **Introduction to module 2- Vector** | **Semester & Section:** | **6TH SEM & A Section** |
| **Github Repository:** | **Dhavala27** |  |  |

|  |
| --- |
| **SESSION DETAILS** |

**DAILY ASSESSMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **14/07/2020** | **Name:** | **Dhavala** |
| **Course:** | **Salseforce** | **USN:** | **4AL17EC027** |
| **Topic:** | * **Platform development Basics** | **Semester & Section:** | **6TH SEM & A Section** |
| **Github Repository:** | **Dhavala27** |  |  |

|  |
| --- |
| **SESSION DETAILS**      **The Power of Metadata**  If you completed the Salesforce Platform Basics module, you probably remember the discussion of Salesforce terminology. Specifically, we defined three important terms: **objects**, **fields**, and **records**. Let’s take that definition one step further.  When you look at data in Salesforce, you might think that you're looking at a user interface sitting on top of a run-of-the-mill relational database. But what you’re actually looking at is an abstraction of the database that’s driven by the platform’s metadata-aware architecture.  In this abstraction, objects are our database tables. The fields on those objects are columns, and records are rows in the database. This analogy is true both for **standard objects** that come with Salesforce by default and **custom objects** that you build yourself.  You can see metadata in action on record detail pages. On this detail page for an account record, you can see field names like Type, Account Number, and Website. These are the metadata that define the structure of your app. The values of each of these fields are our actual data, and in terms of our data model, they aren’t particularly relevant. In short, metadata forms the structure of your org. Whether you’re defining fields, business processes, or something more complex, metadata holds your configuration. The platform then renders your app’s metadata in the user interface along with its associated data.  This **metadata-driven development model** is one of the key differences between developing on the platform and developing outside of Salesforce. Because the platform is metadata-aware, it can auto-generate a significant part of your user experience for you. Things like dialogs, record lists, detail views, and forms that you’d normally have to develop by yourself come for free. You even get all the functionality to create, read, update, and delete (affectionately referred to as CRUD) custom object records in the database. All this prebuilt functionality frees up your development time to work on more sophisticated custom features. Let’s take a look at how the metadata-driven development approach works in action.  **What’s Inside DreamHouse?**  The DreamHouse app is an example of what’s typically referred to as an internal employee productivity app. It’s built using various parts of the Salesforce platform. This diagram gives an overview of the system landscape of the DreamHouse application set. It includes Heroku services and other connected devices, but let’s focus on the core platform piece for the time being.  **No-Code and Low-Code Development**  It can be surprising to hear, but the Salesforce platform encourages you to minimize code. And it’s not because we don’t love code. It’s because the platform’s metadata-driven architecture lets you complete most basic development tasks without ever writing a line.  Salesforce offers a host of tools for point-and-click—or **declarative**—development. Most of these tools require little to no understanding of development principles: no code. Someone who thinks JSON is just missing a letter can construct a robust and complex data model. A person who hears Cron and thinks it’s some kind of sci-fi film can schedule batch jobs.  In fact, someone without any coding knowledge at all can develop entire apps in Salesforce using prebuilt components and point-and-click tools. Here’s an example of Lightning App Builder, which we used to build parts of the DreamHouse app.  For DreamHouse, we’ve created three custom objects that support the app’s core functionality.   * Brokers: information about partner brokers * Properties: photos and information about properties that are on the market * Favorites: properties customers have favorited     The no-code and low-code development capabilities that the Salesforce platform provides means that you, as a developer, can move faster. If you’re the only person at your company developing on Salesforce, you can use the platform’s many declarative tools to build more in less time. If you’re working on a team with non-coders, you can leave the declarative development tasks to them while you double down on more code-intensive projects. |