Lightning component development using LWC\_ Day 1

Logistics

1. Start time is: 09:00 AM and end time is: 05:00 PM (15 mins Tea break and 1 hour Lunch break)
2. Material will be provided as Salesforce EKIT (pdf version) after completion of training
3. Salesforce activities and case study to understand the framework and implement app development process.

Pre-requisites:

1. Developer oriented, so participant should be having Apex programming background.
2. UI technology understanding is an added advantage

Lightning component development using LWC is called DEX602 session in salesforce.

Introduction to Lightning components in salesforce

Salesforce has 2 different UI standards in their applications

1. Classic UI
2. Lightning UI (Lightning experience)

Classic UI:

Is standard UI of salesforce for their classic applications which run in standard web browsers of cloud.

Classic UI has traditional user experience built on HTML based pages. The pages representing classic UI has inconsistent user experience between different standard browsers in market.

Classic UI is not compatible with mobile browsers or mobile apps designed for IOS or Android.

Ex: pages built using Visualforce programming

Lightning UI:

Is upgraded user experience in salesforce where developers can create highly interactive and responsive pages for customer interaction using web browsers or mobile apps while working on salesforce applications.

Advantages of Lightning over Classic:

* Lightning is built on HTML5 which is Schema based markup language better adoptable for web pages built irrespective of Operating systems (IOS or Android). All device compatibility.
* Lightning used “component-based development” where the page is made of components and transactions happening in the component will reload or refresh the specific component leaving the remaining page STATIC. This will enhance performance of the pages, and this is called “Single Page Architecture”
* Transactions happen in Async mode in lightning which helps in rendering the output as continuous stream in the UI and end-user need not wait for the response and work with application in un-interrupted way.
* It uses salesforce pre-defined stylesheets to maintain consistency in application user experience irrespective of which device is used for launching of the application. The stylesheets are referred from “Salesforce Lightning Design System” for branding.

Salesforce Lightning Frameworks for building lightning components

1. Standard Salesforce Lightning framework – AURA framework

Standard and Proprietary framework of salesforce.

It is heavy weight framework where for each component, salesforce provides a collection of files called “component bundle” to be created for developing requirement.

Component bundle is collection of 8 files which are linked together in the framework to support component execution and generate huge HTML5 code when it gets compiles.

* Component
* Controller
* Helper
* Style
* Documentation
* Design
* Renderer
* SVG

Pro – it is time tested framework and mature for developing of applications

Con – developer need to get trained on the framework and component will take little more time to open in the web browser or mobile app.

1. Lightning Web Components (LWC) framework

Recent framework based on Open-Source web technologies in the market. This framework is directly built on Web standards and components will be compiled and generate less amount of code in the backend so that they will be executing faster in web browser or mobile app.

LWC components are compatible with AURA components for complex requirements where developer can embed LWC component into existing AURA component for handling.

(vice-versa is not possible)

LWC components are easy to build and uses less amount of code, hence developers can easily understand and work with framework.

LWC components can be developed offline and deploy into salesforce once the component is built. Developer can use third party editors which gives “content assist” in building of components.

Lightning Application Development in Salesforce

Lightning applications are built using “tool” in salesforce called “App Manager” and applications are made up of “pages” and to build the pages, there is a tool called “lightning app builder”

Lightning Pages:

UI for Salesforce application pages are there are 3 types of lightning pages

1. App Page – single page application with multiple components are created
2. Home Page – Landing page for application to be launched by “HOME” tab
3. Record Page – page that opens when salesforce record is accessed.

Pages are created using components and they are classified into 3 types

1. Standard component – pre-built components
2. Custom component – developer defined components
3. Custom – Managed component – installed components from “AppExchange”

**Custom Component Development using LWC**

Setting up environment for custom component development

1. Download and install Salesforce CLI in the machine

Test the installation success using “sfdx” command in command prompt of machine.

1. Download and install “Visual studio code” IDE from the internet (System Installer – 64bit)

* Enable “salesforce extension pack” and restart the IDE
* Create Project folder for development
* Authorize Salesforce Org into the IDE for deployment

LWC component development

Component can be built in LWC using LWC component bundle.

There are some rules to be followed “strictly” for naming conventions in building components.

1. Component bundle should always begin with lower case and if the name comprises of multiple words, every other word first char should be in CAPS (camel notation)
2. All files in the component bundle of LWC will have same name as folder name and hence the naming conventions need to be strict.
3. Component bundle will have 3 files: HTML, JS and XML (meta) and these files should be deployed together as LWC component.
4. Deployment should be done from the HTML file, which will link up other files to deploy automatically into salesforce.

Component Creation:

In vscode, use command “sfdx:create lightning web component”

Name the component using “naming conventions”

3 files will be created

1. HTML: is the source file for component User Interface. It starts with “template” tag which is given by LWC framework, and it will be replaced with component name when the component renders in the web browser.
2. JS: contains JS class that is used for creation of variables and methods for working with data and binding data to the HTML file elements
3. XML: is meta file used for publishing component into salesforce lightning and helps in providing configs of the component.

After providing requirement implementation, deploy the HTML file which will include all the files in the process of deployment for loading the component into salesforce pages.

Activity:

Create a component for some specific use case in lightning

Pharma Products – component context

Variables:

In lightning components, variables are declared in JS file as “Class Variables” or “Instance variables”

All the variables are default “private” and known as “reactive” variables

Developer can define “public” variables by annotating them with “@api”.

Public variables can have the values set from other components or applications outside the component scope.

Activity:

Implement static analysis on component using “if” condition in program

Will modify pharmaProduct component for adding IF condition.

Collections in Lightning framework:

In lightning, JS class uses Arrays for representing collections in salesforce.

List or Set or Map collection are referred as Arrays for binding data and processing elements from the JS file in lightning components.

Activity:

Create component to represent multiple elements and process them as collection from the UI.

Iteration in lightning components will be done using

For-each

For-item

Properties in the template tag of the HTML file in lightning component.

Custom lightning components can be prepared using pre-defined lightning components provided by salesforce to achieve consistent user experience in developing salesforce applications.

Salesforce provides “Component Library” where pre-built components are available grouped by “namespaces”

Ex:

Lightning-button

Lightning-card

Lightning-pill

Lightning-input etc…

These pre-built components have attributes using which developer can set properties which can be used to customize those components to meet the requirements of custom component.

Building custom components using pre-defined salesforce components is considered as best practice.

Activity:

Create custom component that simulates salesforce standard component using “pre-defined component” from the component library.

Lightning-card

Lightning-button

Lightning-datatable

-------------------------- END of DAY 1--------------------------------------------