**Example :**- Automation Test Scripts for Salesforce Sandbox initial utilization

1.Feature File 2. Step File 3. Page File 4. Runner File

|  |  |  |  |
| --- | --- | --- | --- |
| Feature:  @Example  Scenario outline: Description  Given login to App  Then Verify header  Example: | public Class StepDef {  Page objPage=new Page();  @Given (“login to App”)  public void loginToApp () {  }  }  @Then (“Verify header”)  public verifyHeader () {  driver.findElement(objPage.header);  } | public Class Page {  public By header= By.xpath(“//span(@title=’header’);  } | CucumberOptions(features = "src/test/resources/features", glue = { "com.cognizant.steps" }, tags = {  "@Example"}, strict = true, monochrome = true, plugin = { "pretty",  "pretty:target/cucumber-report/pretty.txt", "html:target/cucumber-report",  "json:target/cucumber-report/cucumber.json", "junit:target/cucumber-report/cucumber-junitreport.xml",  "com.aventstack.extentreports.cucumber.adapter.ExtentCucumberAdapter:" } |

1. **Feature File:** It is a simple feature file contains login and logout verification, selecting dropdown bar, app launcher, global search bar.

**Tip:** If you click on any Given, When, Then, And lines while holding CTRL + Click , it will take you to the related Step in the Step Class.

Feature: Verifying dropdown values in home page, app launcher, Global search bar, login & logout functionalities.  
  
 @HomePage   
 Scenario Outline: Verify the fields which is present in-home page, app launcher, Global search bar, login & logout functionalities

Given Salesforce Application Open  
 When Enter the Salesforce Application Credentials for <Role>  
 Then Salesforce Application Login verification

Then Verify global search box is displayed  
 When Salesforce Application Click dropdown tab and select Item <itemName>  
 When Salesforce Application Logout  
 Then Salesforce Application Logout verification  
  
 Examples:  
 | **Role** | **itemName** |   
 | **HelpDesk Operator** | **Home** |

1. **Step Class**: - It contains the logics and actions which was written in java/ JavaScript. These all interlink to each other with Feature file, Step class, Page class, Runner file.

* Here we will create objects for the Page classes and call the By data-type methods which consists X-Path’s in it.

@Given("Salesforce Application Open")  
public void salesforce\_application\_open() {  
 driver.get(properties.getProperty("ApplicationUrl"));  
 Capabilities cap = ((RemoteWebDriver) driver).getCapabilities();  
 String browserName = cap.getBrowserName().toLowerCase();  
 String osName = cap.getPlatform().toString();  
 String Version = cap.getVersion().toString();  
 addStepLog("Browser name:" + browserName + ", Browser version:" + Version + ", OS name:" + osName);  
 addStepLog("------> Enters Application Url in browser");  
}

* Here the Application URL and Usernames will take from properties file. Below is the example for Properties file

# Framework settings file  
  
ProjectName=   
  
#Default Settings - To Avoid Null Pointer within the Framework  
DateFormatString=dd-MMM-yyyy hh:mm:ss a  
  
#Capture Every Execution Report in Seperate Folder  
SaveReports=True  
  
#Salesforce workbench url  
workbenchUrl = https://workbench.developerforce.com/login.php  
  
# User-defined Settings  
ApplicationUrl=https://test.salesforce.com/  
AppURL=   
UserName=   
Password=   
OpasSupportUserName=   
OpasSupportPassword=   
AssetEngineerUsername=   
AssetEngineerPassword=  
SurveyorSupervisorUsername=   
SurveyorSupervisorPassword=   
#Delivery Manager Credentials  
DeliveryManagerUsername=   
DeliveryManagerPassword=   
LizDeliveryManagerUsername=   
LizDeliveryManagerPassword=   
  
RizwanaUsernameAssetEngineer=   
RizwanaPasswordAssetEngineer=   
ConditionalSurveyorUsername =   
ConditionalSurveyorPassword =   
TechnicalAuthorityUsername=   
TechnicalAuthorityPassword=   
HelpdeskOperatorUsername =   
HelpdeskOperatorPassword =   
SystemAdminUsername=   
SystemAdminPassword=

@When("Enter the Salesforce Application Credentials for (.\*)")  
public void enter\_the\_salesforce\_application\_credentials\_something(String userrole) {  
 switch (userrole.toLowerCase()) {  
 case "opassupport":  
 case "opas":  
 case "opas support":  
 driver.findElement(objSalesforceLoginPage.txtUsername).sendKeys(properties.getProperty("UserName"));  
 driver.findElement(objSalesforceLoginPage.txtPassword).sendKeys(properties.getProperty("Password"));  
 break;  
 case "assetengineer":  
 case "asset engineer":  
 case "nr asset engineer":  
 driver.findElement(objSalesforceLoginPage.txtUsername).sendKeys(properties.getProperty("AssetEngineerUsername"));  
 driver.findElement(objSalesforceLoginPage.txtPassword).sendKeys(properties.getProperty("AssetEngineerPassword"));  
 break;  
 case "surveyorsupervisor":  
 case "surveyor supervisor":  
 driver.findElement(objSalesforceLoginPage.txtUsername).sendKeys(properties.getProperty("SurveyorSupervisorUsername"));  
 driver.findElement(objSalesforceLoginPage.txtPassword).sendKeys(properties.getProperty("SurveyorSupervisorPassword"));  
 break;  
 case "workdelivery manager":  
 case "workdelivery":  
 case "deliverymanager"://delivery team  
 case "delivery manager"://Works Delivery Manager  
 driver.findElement(objSalesforceLoginPage.txtUsername).sendKeys(properties.getProperty("DeliveryManagerUsername"));  
 driver.findElement(objSalesforceLoginPage.txtPassword).sendKeys(properties.getProperty("DeliveryManagerPassword"));  
 break;  
 case "conditionsurveyor":  
 case "conditional surveyor":  
 case "surveyor"://Conditional Surveyor  
 case "technician":  
 driver.findElement(objSalesforceLoginPage.txtUsername).sendKeys(properties.getProperty("ConditionalSurveyorUsername"));  
 driver.findElement(objSalesforceLoginPage.txtPassword).sendKeys(properties.getProperty("ConditionalSurveyorPassword"));  
 break;  
 case "technical authority":  
 case "technicalauthority":  
 driver.findElement(objSalesforceLoginPage.txtUsername).sendKeys(properties.getProperty("TechnicalAuthorityUsername"));  
 driver.findElement(objSalesforceLoginPage.txtPassword).sendKeys(properties.getProperty("TechnicalAuthorityPassword"));  
 break;  
 case "helpdesk operator":  
 case "helpdeskoperator":  
 driver.findElement(objSalesforceLoginPage.txtUsername).sendKeys(properties.getProperty("HelpdeskOperatorUsername"));  
 driver.findElement(objSalesforceLoginPage.txtPassword).sendKeys(properties.getProperty("HelpdeskOperatorPassword"));  
 break;  
 case "Contractor Manager":  
 case "ContractorManager":  
 driver.findElement(objSalesforceLoginPage.txtUsername).sendKeys(properties.getProperty("ContractorManagerUsername"));  
 driver.findElement(objSalesforceLoginPage.txtPassword).sendKeys(properties.getProperty("ContractorManagerPassword"));  
 case "systemadmin":  
 case "system admin":  
 driver.findElement(objSalesforceLoginPage.txtUsername).sendKeys(properties.getProperty("SystemAdminUsername"));  
 driver.findElement(objSalesforceLoginPage.txtPassword).sendKeys(properties.getProperty("SystemAdminPassword"));  
 break;  
 case "opasmangala":  
 driver.findElement(objSalesforceLoginPage.txtUsername).sendKeys(properties.getProperty("OpasSupportUserName"));  
 driver.findElement(objSalesforceLoginPage.txtPassword).sendKeys(properties.getProperty("OpasSupportPassword"));  
 break;  
 case "helpdesk operator2":  
 case "helpdeskoperator2":  
 driver.findElement(objSalesforceLoginPage.txtUsername).sendKeys(properties.getProperty("HelpdeskOperatorUsername2"));  
 driver.findElement(objSalesforceLoginPage.txtPassword).sendKeys(properties.getProperty("HelpdeskOperatorPassword2"));  
 break;  
 case "liz dm":// liz Works Delivery Manager  
 driver.findElement(objSalesforceLoginPage.txtUsername).sendKeys(properties.getProperty("LizDeliveryManagerUsername"));  
 driver.findElement(objSalesforceLoginPage.txtPassword).sendKeys(properties.getProperty("LizDeliveryManagerPassword"));  
 break;  
 case "george commercial manager":// george commercial manager  
 driver.findElement(objSalesforceLoginPage.txtUsername).sendKeys(properties.getProperty("GeorgeCommercialManagerUsername"));  
 driver.findElement(objSalesforceLoginPage.txtPassword).sendKeys(properties.getProperty("GeorgeCommercialManagerPassword"));  
 break;  
 case "asset engineer dev":  
 driver.findElement(objSalesforceLoginPage.txtUsername).sendKeys(properties.getProperty("AssetEngineerDevUserName"));  
 driver.findElement(objSalesforceLoginPage.txtPassword).sendKeys(properties.getProperty("AssetEngineerDevPassword"));  
 break;  
 case "delivery manager dev":  
 driver.findElement(objSalesforceLoginPage.txtUsername).sendKeys(properties.getProperty("DeliveryManagerDevUserName"));  
 driver.findElement(objSalesforceLoginPage.txtPassword).sendKeys(properties.getProperty("DeliveryManagerDevPassword"));  
 break;  
 case "help desk operator dev":  
 driver.findElement(objSalesforceLoginPage.txtUsername).sendKeys(properties.getProperty("HelpDeskOperatorDevUserName"));  
 driver.findElement(objSalesforceLoginPage.txtPassword).sendKeys(properties.getProperty("HelpDeskOperatorDevPassword"));  
 break;  
 case "opas support dev":  
 driver.findElement(objSalesforceLoginPage.txtUsername).sendKeys(properties.getProperty("OpasSupportDevUserName"));  
 driver.findElement(objSalesforceLoginPage.txtPassword).sendKeys(properties.getProperty("OpasSupportDevPassword"));  
 break;  
 case "commercial manager dev":  
 driver.findElement(objSalesforceLoginPage.txtUsername).sendKeys(properties.getProperty("CommercialManagerDevUserName"));  
 driver.findElement(objSalesforceLoginPage.txtPassword).sendKeys(properties.getProperty("CommercialManagerDevPassword"));  
 break;  
 case "rizwana asset engineer":  
 driver.findElement(objSalesforceLoginPage.txtUsername).sendKeys(properties.getProperty("RizwanaUsernameAssetEngineer"));  
 driver.findElement(objSalesforceLoginPage.txtPassword).sendKeys(properties.getProperty("RizwanaPasswordAssetEngineer"));  
 break;  
 default:  
 System.*out*.println("Please provide valid User Role");  
 }  
 try {  
 driver.findElement(objSalesforceLoginPage.btnLogin).click();  
 }catch(WebDriverException e){  
 addStepLog("caught cannot determine loading status");  
 }  
 waitFor(2000);  
  
 addStepLog("-------> Logged in as an " + userrole + "user");  
}

@Then("Salesforce Application Login verification")  
public void salesforce\_application\_login\_verification() {  
 if(objectExistsCheck(objSalesforceLoginPage.btnTryAgain)){  
 attachScreenshotForWeb();  
 JSscriptclick(driver.findElement(objSalesforceLoginPage.btnTryAgain));  
 driver.navigate().refresh();  
 }  
 if (objectExistsCheck(objSalesforceLoginPage.popupLogin)) {  
 attachScreenshotForWeb();  
 JSscriptclick(driver.findElement(objSalesforceLoginPage.popupLogin));  
 driver.navigate().refresh();  
 }  
 waitForPageLoad(20);  
 if(objectExistsCheck(objSalesforceLoginPage.btnTryAgain)){  
 attachScreenshotForWeb();  
 JSscriptclick(driver.findElement(objSalesforceLoginPage.btnTryAgain));  
 driver.navigate().refresh();  
 }  
 try {  
 waitUntilElementLocated(objSalesforceHomePage.setUP, 60);  
 }catch(Exception e){  
 driver.navigate().refresh();  
 }  
 if (!objectExists(objSalesforceHomePage.NetworkRailAppTitle)) {  
 openTheNetworkRailApp();  
 }  
 addStepLog("----> Application page is launched successfully after login");  
}

@Then ("Verify global search box is displayed")  
public void verifyGlobalSearch() {  
 // pageRefresh();  
 waitUntilElementVisible(objSalesforceHomePage.globalSearchBox, 30);  
 *assertTrue*(objectExists(objSalesforceHomePage.globalSearchBox), "Global Search displayed");  
}

@When("Navigate and Open (.\*) tab in Salesforce Application")  
public void navigateAndOpenInSalesforce(String name) {  
 {  
 addStepLog("Expected - The User should be navigate to " + name + " tab");  
 waitFor(3000);  
 waitUntilElementVisible(objSalesforceHomePage.dropdownButtonOnMainTab, 30);  
 if (objectExistsCheck(objSalesforceLoginPage.btnTryAgain)) {  
 attachScreenshotForWeb();  
 JSscriptclick(driver.findElement(objSalesforceLoginPage.btnTryAgain));  
 driver.navigate().refresh();  
 }  
 if (objectExistsCheck(objSalesforceLoginPage.popupLogin)) {  
 attachScreenshotForWeb();  
 JSscriptclick(driver.findElement(objSalesforceLoginPage.popupLogin));  
 driver.navigate().refresh();  
 }  
 waitUntilElementVisible(objSalesforceHomePage.dropdownButtonOnMainTab, 15);  
 attachScreenshotForWeb();  
 driver.findElement(objSalesforceHomePage.dropdownButtonOnMainTab).click();  
 waitFor(3000);  
 if (objectExists(objSalesforceHomePage.menuItemsInMainTab(name))) {  
 driver.findElement(objSalesforceHomePage.menuItemsInMainTab(name)).click();  
 } else {  
 driver.findElement(objSalesforceHomePage.dropdownButtonOnMainTab).click();  
 waitFor(4000);  
 waitUntilElementLocated(objSalesforceAssetHierarchyPage.AssetHierarchyAL, 10);  
 waitUntilElementClickable(objSalesforceAssetHierarchyPage.AssetHierarchyAL, 10);  
 driver.findElement(objSalesforceAssetHierarchyPage.AssetHierarchyAL).click();  
 waitUntilElementLocated(objSalesforceAssetHierarchyPage.linkViewAll, 10);  
 driver.findElement(objSalesforceAssetHierarchyPage.linkViewAll).click();  
 if (!objectExists(objSalesforceAssetHierarchyPage.linkAssetHierarchy)) {  
 waitFor(2000);  
 driver.findElement(objSalesforceAssetHierarchyPage.btnClose).click();  
 driver.findElement(objSalesforceAssetHierarchyPage.AssetHierarchyAL).click();  
 waitUntilElementLocated(objSalesforceAssetHierarchyPage.linkViewAll, 10);  
 driver.findElement(objSalesforceAssetHierarchyPage.linkViewAll).click();  
 }  
 waitUntilElementLocated(objSalesforceNRYears.searchItem, 20);  
 driver.findElement(objSalesforceNRYears.searchItem).click();  
 driver.findElement(objSalesforceNRYears.searchItem).sendKeys(name);  
 waitUntilElementLocated(objSalesforceNRYears.viewAllLinkItem(name), 20);  
 attachScreenshotForWeb();  
 JSscriptclick(driver.findElement(objSalesforceNRYears.viewAllLinkItem(name)));  
 waitFor(1000);  
 }  
 do {  
 waitFor(2000);  
 if (objectExistsCheck(objSalesforceAssetsPage.closeOtherTabs(name))) {  
 JSscriptclick(driver.findElement(objSalesforceAssetsPage.closeOtherTabs(name)));  
 }  
 } while (objectExists(objSalesforceAssetsPage.closeOtherTabs(name)));  
  
 addStepLog("Expected - The User is navigated to " + name + " tab");  
 }  
}

@When("Salesforce Application Logout")  
public void salesforce\_application\_logout() {  
 waitFor(2000);  
 if(!objectExists(objSalesforceHomePage.viewProfile))  
 driver.switchTo().defaultContent();  
 if (!objectExists(objSalesforceHomePage.viewProfile)) {  
 driver.navigate().refresh();  
 waitFor(15000);  
 }  
 waitUntilElementLocated(objSalesforceHomePage.viewProfile, 30);  
 JSscriptclick(driver.findElement(objSalesforceHomePage.viewProfile));  
 if (objectExists(objSalesforceHomePage.SalesforceProfile)) {  
 //do nothing  
 } else {  
 waitUntilElementLocated(objSalesforceHomePage.viewProfile, 120);  
 waitUntilElementClickable(objSalesforceHomePage.viewProfile, 20);  
 JSscriptclick(driver.findElement(objSalesforceHomePage.viewProfile));  
 waitFor(2000);  
 }  
 waitUntilElementLocated(objSalesforceHomePage.SalesforceProfile, 20);  
 waitUntilElementLocated(objSalesforceHomePage.SalesforceDomain, 20);  
 waitUntilElementLocated(objSalesforceHomePage.SalesforceSettings, 20);  
 JSscriptclick(driver.findElement(objSalesforceHomePage.Salesforcelogout));  
 addStepLog("----> Logout operation is performed");  
}

@Then("Salesforce Application Logout verification")  
public void salesforce\_application\_logout\_verification() {  
 addStepLog("------> Expected : User should able to view login screen");  
 wait.until(ExpectedConditions.*visibilityOfElementLocated*(objSalesforceLoginPage.txtUsername));  
 waitFor(2000);  
 addStepLog("----> Actual : User able to view login screen");  
}

1. **Page Class:** In Page Class we will store X-paths in the methods with data type as By

public class SalesforceLoginPage {  
  
 //Text box  
 public final By txtUsername = By.*xpath*("//input[@type='email']");  
 public final By txtPassword= By.*xpath*("//input[@type='password']");  
  
 //Button  
 public final By btnLogin= By.*xpath*("//input[@type='submit' and @value='Log In to Sandbox']");  
  
 //Checkbox  
 public final By cbxRememberMe= By.*xpath*("//label[text()='Remember me']//preceding-sibling::input[@type='checkbox']");  
 public final By popupLogin= By.*xpath*("//button[text()='Log In']");  
  
 public By btnTryAgain=By.*xpath*("//\*[text()='Try Again']");

}

public class SalesforceHomePage {  
  
 // Text box, button, links, labels, dropdown list  
  
 public By selectAssetName(String Item)  
 {  
 return By.*xpath*("//a[text()='"+Item+"']");  
 }  
  
 public final By HCEAvailable = By.*xpath*("(//\*[text()='HCE']//following::div)[1]");  
  
  
 //setup  
 public final By setUP= By.*xpath*("//\*[@data-key='setup']");  
  
 public final By setUPoption= By.*xpath*("//li[@class='slds-dropdown\_\_item related-app uiMenuItem onesetupSetupMenuItem']");  
  
 //ViewProfilePage  
 public final By viewProfile= By.*xpath*("//span[@class='uiImage' or contains(@class,'photo')]/img[@class='icon noicon' or contains(@class,'user-profile')]");  
  
 //UserName  
 public final By UserName=By.*xpath*("//h1[@class='profile-card-name']");  
  
 //ViewProfileSetting  
 public final By viewProfileSetting= By.*xpath*("//\*[@class='profile-card-toplinks']/a[1]");  
  
 //Network Rail logo  
 public final By NetworkRail= By.*xpath*("//span[@title='Network Rail']");  
  
 //user  
 public final By Salesforceuser= By.*xpath*("//img[@title='User']");  
  
 //logout  
 public final By Salesforcelogout= By.*xpath*("//a[text()='Log Out']");  
  
 //Profile  
 public final By SalesforceProfile= By.*xpath*("//\*[@class='profile-link-label' and text()!='Settings']");  
  
 //Domain  
 public final By SalesforceDomain= By.*xpath*("//\*[@class='profile-card-domain']");  
  
 //Settings  
 public final By SalesforceSettings= By.*xpath*("//\*[@class='profile-link-label' and text()='Settings']");  
  
 //link  
 public final By tabHome= By.*xpath*("//a[@title='Home']");  
  
 //Handling span  
 public final By lnkOpenCases= By.*xpath*("//span[text()='All Open Cases']");  
  
 //Handling li  
 public final By lnkFirstOpenCase= By.*xpath*("//li[1]//span[text()=' New']");  
  
 //Application Selector  
 public final By iconAppSelector = By.*xpath*("//div[@role='navigation']");  
 public final By tbxAppSearchBox = By.*xpath*("//input[@type='search' and @placeholder='Search apps and items...']");  
 public final By tabSalesApp = By.*xpath*("//a[@data-label='Sales']");  
  
 //Case by origin chart  
 public final By btnRefresh = By.*xpath*("//lightning-icon[@class='slds-icon-utility-refresh slds-icon\_container']");  
 public final By lnkViewReport = By.*xpath*("//button//span[text()='View Report']");  
  
 //Tile heading  
 public final By lblCaseByOrigin = By.*xpath*("//span[@title='Cases by Origin and Open Status']");  
 public final By lblTodaysTask = By.*xpath*("//span[@title='Today’s Tasks']");  
  
 public final By globalSearchBox = By.*xpath*("//header[@id='oneHeader']//button[contains(@class,'search')]");  
 public final By DeleteButtonAfterSearchingAssetFromGlobalSearch = By.*xpath*("(//div[text()='Delete'])[1]");  
 public final By DeleteButtonInsidePopUp = By.*xpath*("//span[text()='Delete']");  
 public By firstAvailableItemActiveTab=By.*xpath*("(//div[contains(@class,'maximized active')]//tbody//a)[1]");  
 public By firstAvailableItemMainTab=By.*xpath*("(//div[contains(@class,'maximized oneContent active')]//tbody//a)[1]");  
 public By listViewBtn(String btnName) {  
 return By.*xpath*("//div[contains(@class,'maximized active') or contains(@class,'maximized oneContent active')]//a[@title='"+btnName+"']");  
 }  
 public By DdlEnteredItemName(String Item)  
 {  
 return By.*xpath*("//mark[text()='"+Item+"']");  
 }  
// public final By DeleteButtonInsideDropDown = By.xpath("//div[text()='Delete']");  
  
 public By afterSearchResultDropdownBtn (String name){  
 return By.*xpath*("(//a[text()='"+name+"']/following::a[contains(@class,'rowActionsPlaceHolder') and @aria-expanded='false'])[1]");  
 }  
  
 public final By dropdownButtonOnMainTab=By.*xpath*("//div[contains(@class,'oneAppNavMenu')]");  
 public final By NetworkRailAppTitle=By.*xpath*("//span[@title='Network Rail']");  
 public final By NetworkRailAppOption=By.*xpath*("//p[text()='Network Rail']");  
 public final By assetEventNewButton=By.*xpath*("//div[@title='New']");  
 public final By globalSearchBox2=By.*xpath*("//input[@type='search' and contains(@placeholder,'more')]|//input[@type='search' and contains(@placeholder,'Search...')]");  
 public By closeButtonsTab=By.*xpath*("//div/button[contains(@title,'Close ')]");  
 public final By uniqueIdentifierError = By.*xpath*("//label[text()='Unique Identifier']/parent::td/following-sibling::td/descendant::div[contains(text(),' Duplicate value on record:')]");  
 public final By userHeader = By.*xpath*("//h1/span[text()='Users']");  
 public final By navigateBackToMaintenancePlan = By.*xpath*("//span[@title='MaintenancePlan']/parent::a[@class='tabHeader slds-tabs--default\_\_link slds-p-right--small slds-grow ']/span[@class='title slds-truncate']");  
 public By unSavedChangesInTab=By.*xpath*("//abbr[@title='Unsaved Changes']");  
 public By cssOfSearchResultPage=By.*cssSelector*("[data-aura-class='uiScroller forceSearchResultsMultiScope']");  
 public By activeChildTabCloseButtonUnderAllMain=By.*xpath*("(//div[contains(@class,'sub-tabs')]//li[contains(@class,'active')]//button[contains(@title,'Close')])");  
  
 //Search value suggestion dropdown  
 public By suggestionValues(String suggestValues){  
 return By.*xpath*("(//span[@title='"+suggestValues+"']/parent::div)[1]");  
 }  
  
 public final By suggestionValueAsset(String value){  
 return By.*xpath*("(//span[@title='"+value+"']/parent::div//span[contains(text(),'Asset')]//ancestor::div[contains(@class,'result-item')])[1]");  
 }  
 public By showMoreResultValue(String suggestValues){  
 return By.*xpath*("(//div[contains(@class,'maximized active')]//a[@title='"+suggestValues+"'])[1]");  
 }  
  
 public By afterSearchResult(String name){  
 return By.*xpath*("(//a[text()='"+name+"'])[1]");  
 }  
 public final By clickShowMoreResult=By.*xpath*("//lightning-formatted-rich-text[contains(@class,'primary slds-truncate')]/span[contains(text(),'Show more results')]");  
  
 public By suggestionValuesOfAsset(String value) {  
 return By.*xpath*("//span[@title='"+value+"']/following-sibling::div//span[text()='Asset']/ancestor::div[@aria-haspopup]");  
 }  
  
 //Verify expected page tab  
 public By expectedTab(String expectedPage)  
 {  
 return By.*xpath*("(//span[contains(@class,'title') and contains(text(),'"+expectedPage+"')])[1]");  
 }  
 //search value with search icon  
 public final By searchOptionWithIcon=By.*xpath*("//a[contains(@class,'SEARCH\_OPTION')]");  
 public final By btnEdit= By.*xpath*("(//input[@value=' Edit '])[1]");  
 public final By iframeXpath= By.*xpath*("//div[@class='content iframe-parent']/descendant::iframe[1]");  
 public final By txtEditEmail = By.*xpath*("//\*[@id='Email']");  
 public By inputUniqueIdentifier = By.*xpath*("//\*[text()='Unique Identifier']/parent::td/following-sibling::\*/input");  
 public final By optnAdvancedUser= By.*xpath*("//div[@title='Advanced User Details']/descendant::a[1]");  
 public final By btnClickSave= By.*xpath*("(//input[@value=' Save '])[1]");  
 public final By updateRolefail= By.*xpath*("//div[@class='pbError']");  
 public By clickAlphabet(String letter){  
 return By.*xpath*("(//span[@class='listItemPad' and text()='"+letter+"'])[1]");  
 }  
  
  
 public By menuItemsInMainTab(String itemName) {  
 return By.*xpath*("//a[@class='menuItem']//span[text()='"+itemName+"']");  
 }  
  
 public By fieldHeaders(String name) {  
 return By.*xpath*("//h2/a[text()='"+name+"']");  
 }  
  
 public By sectionDataLink(String fieldName) {  
 return By.*xpath*("(//a[text()='"+fieldName+"']/ancestor::div[contains(@class,'SearchResultsGridHeader')]/following-sibling::div/descendant::a[@data-refid='recordId'])[1]");  
 }  
  
 public By currentTabCloseButton(int count) {  
 return By.*xpath*("(//div/button[contains(@title,'Close ')])["+count+"]");  
 }  
  
 public By listViewLoader = By.*xpath*("//div[@class='slds-spinner\_container slds-grid']");  
  
 public By headerTitle(String headerName) {  
 return By.*xpath*("//h1/div[text()='"+headerName+"']");  
 }  
 // public By ssmScore = By.xpath("(//span[text()='SSM Score'])[1]");//\*[text()='SSM Score']  
 public By ssmScore = By.*xpath*("//\*[text()='SSM Score']");  
 public By validatingFields = By.*xpath*("(//\*[text()='SSM Score']/parent::div//span)[1]");  
 public By colour = By.*xpath*("//\*[text()='SSM Score']/parent::div//span/img");  
 public By masterWorkbankClick = By.*xpath*("//span[text()='Master Work Bank']/parent::div/following-sibling::div/span/slot/force-lookup/div/records-hoverable-link/div/a");  
 public By masterWorkbankStatus = By.*xpath*("(//div/\*[text()='Status']/parent::\*/following-sibling::\*/span)[2]");  
 public final By QualityReport = By.*xpath*("//a[text()='QC Report']");  
  
 public By notificationBtn = By.*xpath*("//button[contains(@class,'notification')]");  
  
 public By notificationPanelHeader = By.*xpath*("//div[contains(@class,'NotificationsPanel')]//h2[text()='Notifications']");  
  
 public By responsiblePesronOptions(String name){  
 return By.*xpath*("//h2[text()='New Responsible Person']/following::span[text()='"+name+" Responsible Person']");  
 }  
  
  
 public By notificationMessageForRyan = By.*xpath*("((//span[contains(text(),'Ryan Smith (Manager) mentioned you in a post')])[1]/following-sibling::span)[1]");  
  
 public By closeNotifications = By.*xpath*("//button[@title='Close Notifications']");  
  
 public By tabsInsidePage = By.*xpath*("//\*[@slot='main' or contains(@class,'region-main') or @class='col main-col ']//h2/following-sibling::\*//\*[@role='tab']");  
  
 public By tabsInsidePageFirstElement = By.*xpath*("(//\*[@slot='main' or contains(@class,'region-main') or @class='col main-col ']//h2/following-sibling::\*//\*[@role='tab'])[1]");  
  
 public By selectedTabVerify = By.*xpath*("//\*[@slot='main' or contains(@class,'region-main') or @class='col main-col ']//h2/following-sibling::\*//\*[@role='tab' and @aria-selected='true']");  
  
 public By selectMainTab(String tabName){  
 return By.*xpath*("//div[contains(@class,'navCenter')]//a[@title='"+tabName+"']");  
 }  
  
 public By tabsInsidePageListItem = By.*xpath*("//\*[@slot='main' or contains(@class,'region-main') or @class='col main-col ']//h2/following-sibling::\*//\*[@role='tab']/parent::li");  
  
 public By moreBtnInTabs = By.*xpath*("//\*[@slot='main' or contains(@class,'region-main') or @class='col main-col ']//\*[@role='button' and text()='More']");  
  
 public By tabInsidePage(String tabName){  
 return By.*xpath*("//\*[@role='button' and text()='More']//ancestor::li//div[contains(@class,'popup')]//li/a[text()='"+tabName+"']");  
 }  
  
 public final By globalSearchBoxBtn = By.*xpath*("//button[contains(text(),'Search') and contains(@class,'search-button')]");  
  
 public By activeTabCloseButton = By.*xpath*("//li[contains(@class,'item\_tab slds-is-active active')]//button[contains(@title,'Close')]");  
  
 public By activeChildTabCloseButton = By.*xpath*("(//button[contains(@title,'Close')])[2]");  
  
 public By leftChildActiveTab=By.*xpath*("(//div[contains(@class,'left')]/button[contains(@title,'Close')])[2]");  
 public By SiteNameInWorkorderPage=By.*xpath*("//span[@title='Asset']/following-sibling::div/div");  
  
 public By nextBtn=By.*xpath*("//span[text()='Next']");  
 public By searchListInput(String objectName){  
 return By.*xpath*("//input[@placeholder='Search this list...' and contains(@name,'"+objectName+"')]");  
 }  
  
 public By clickLinkBtnAfterSearch(String text){  
 return By.*xpath*("//a[text()='"+text+"']");  
 }  
  
 public By searchButton=By.*xpath*("//span[text()='Asset Wise Documents']/following::button[text()='Search']");  
}

1. **Runner File:** Here in runner class CucumberOptions will give feasibility to run the test cases with tags and run output in a readable way and will generate a cucumber report.

@CucumberOptions(features = "src/test/resources/features", glue = { "com.cognizant.steps" }, tags = {  
 "@E2E\_Sanity\_Check2"}, strict = true, monochrome = true, plugin = { "pretty",  
 "pretty:target/cucumber-report/pretty.txt", "html:target/cucumber-report",  
 "json:target/cucumber-report/cucumber.json", "junit:target/cucumber-report/cucumber-junitreport.xml",  
 "com.aventstack.extentreports.cucumber.adapter.ExtentCucumberAdapter:" })  
  
public class RunnerWebApp extends AbstractTestNGCucumberTests {  
  
 */\*\*\*  
 \* Please enable parallel = true for executing scenarios in Parallel Also  
 \* number of Parallel Threads can be controlled in suite-xml file with  
 \* parameter data-provider-thread-count="1"  
 \*/* @Override  
 @DataProvider(parallel = false)  
 public Object[][] scenarios() {  
 return super.scenarios();  
 }  
}

Here the Test Results for the above Test case

**Feature**: Verifying dropdown values in home page, app launcher, Global search bar, login & logout functionalities.

@HomePage

**Scenario Outline**: Verify the fields which is present in-home page, app launcher, Global search bar, login & logout functionalities

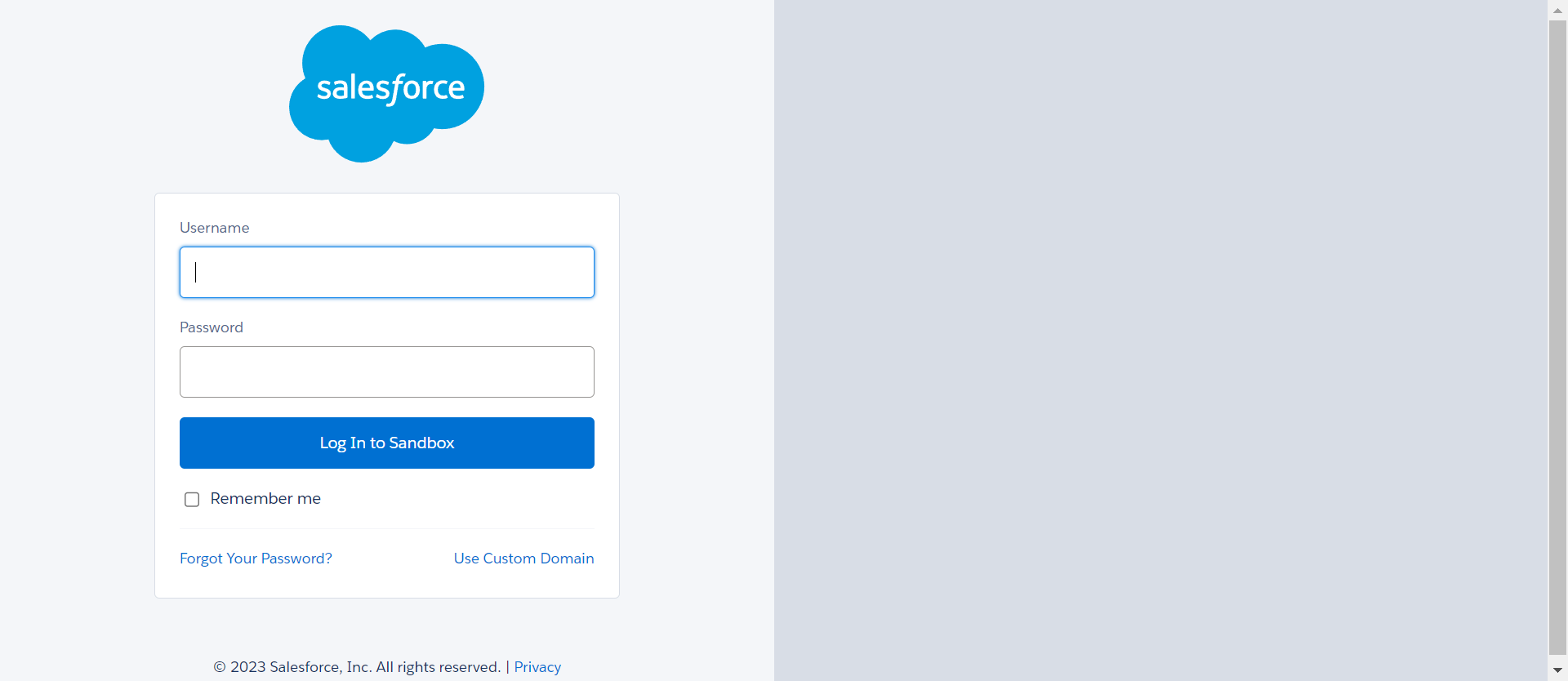
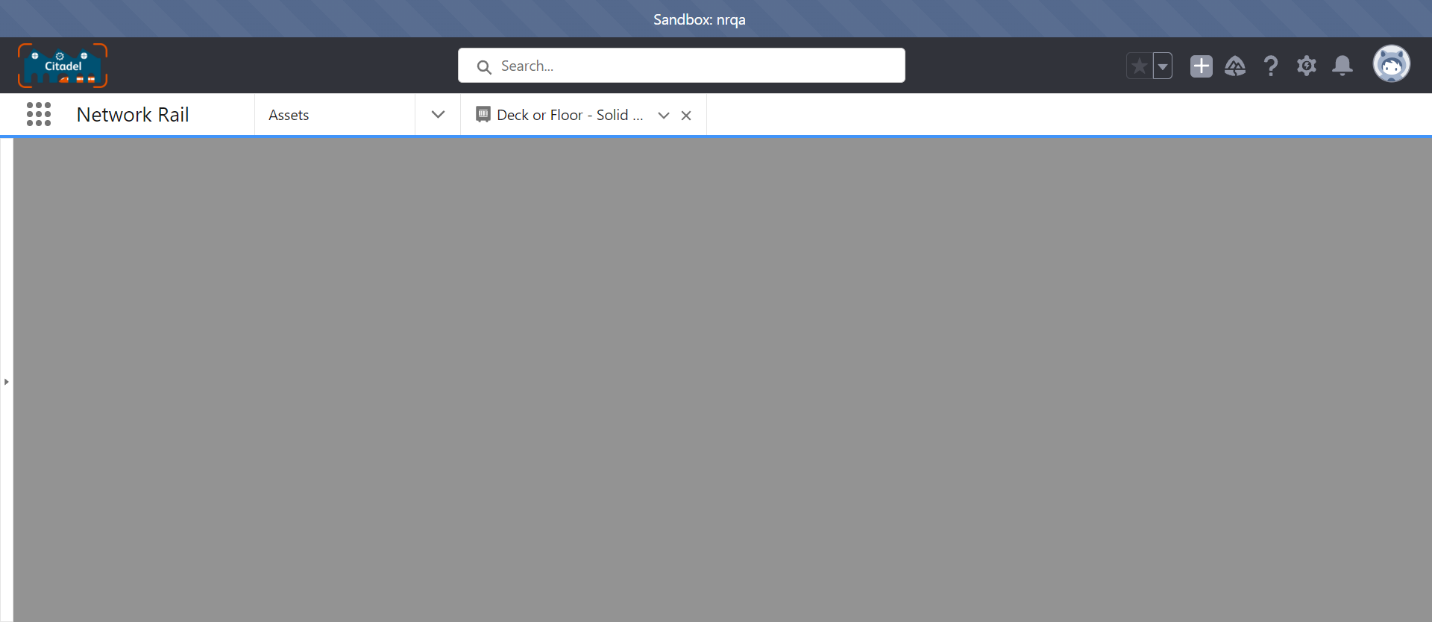
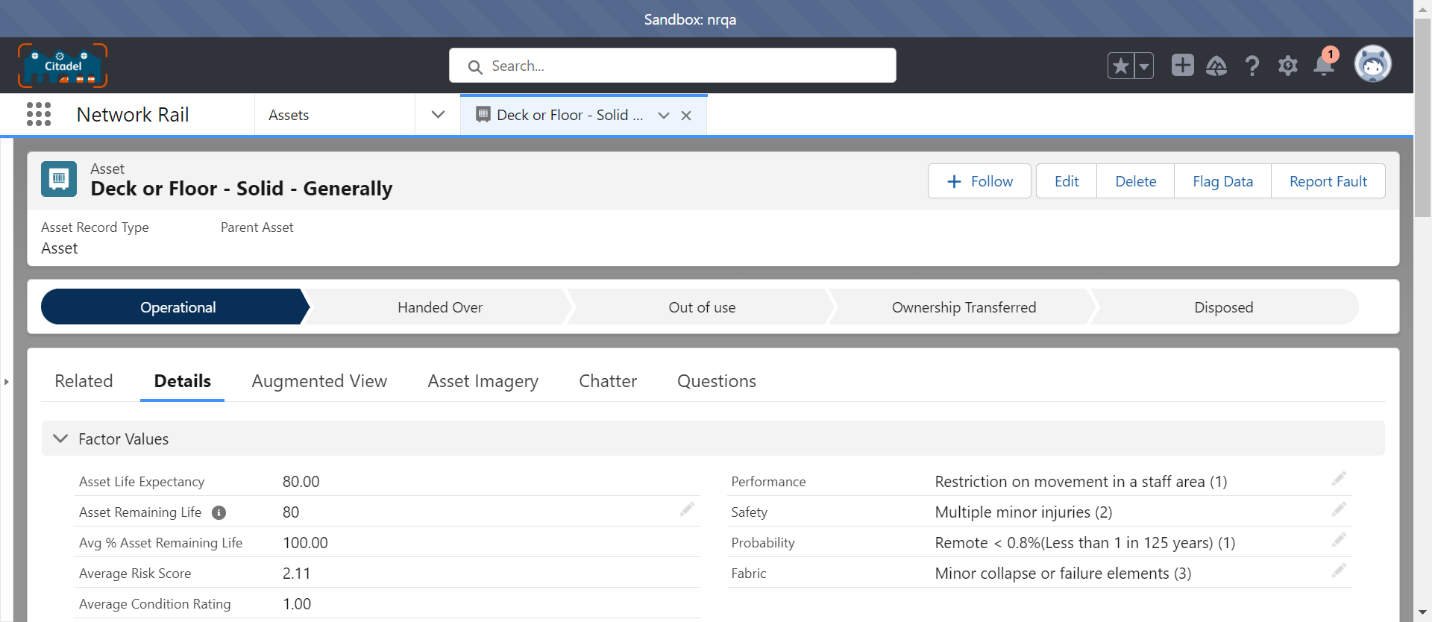
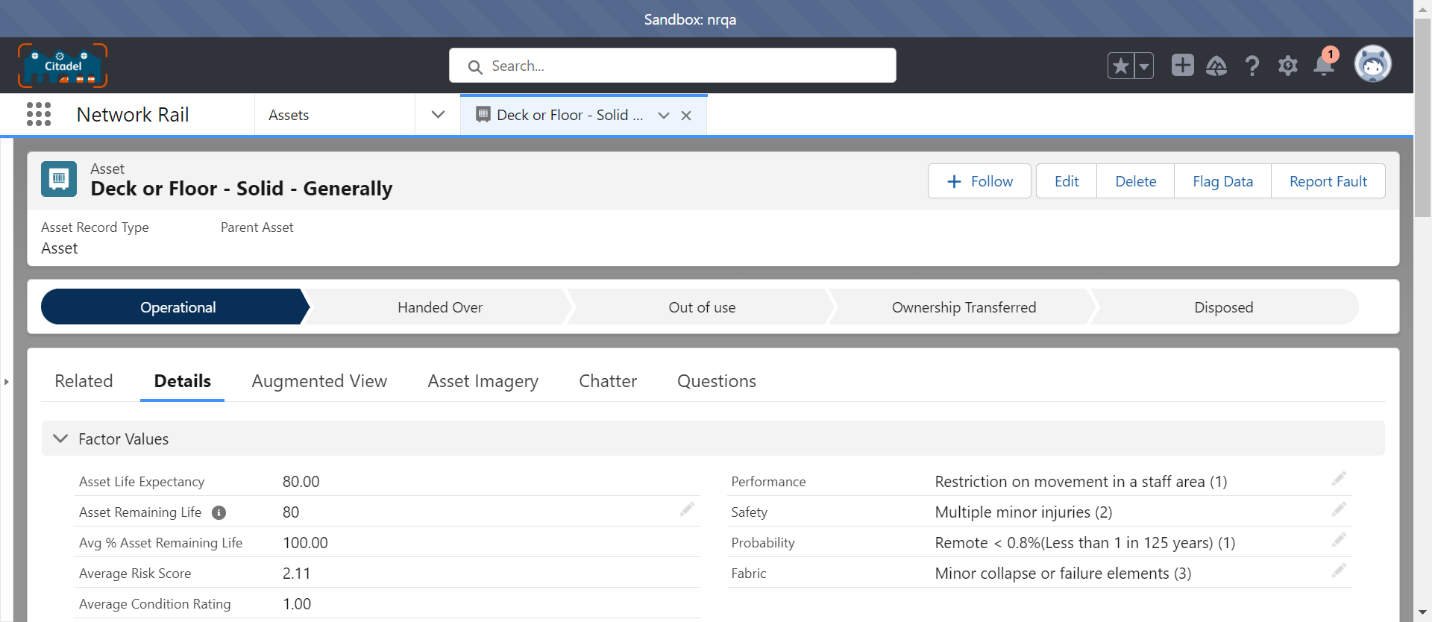
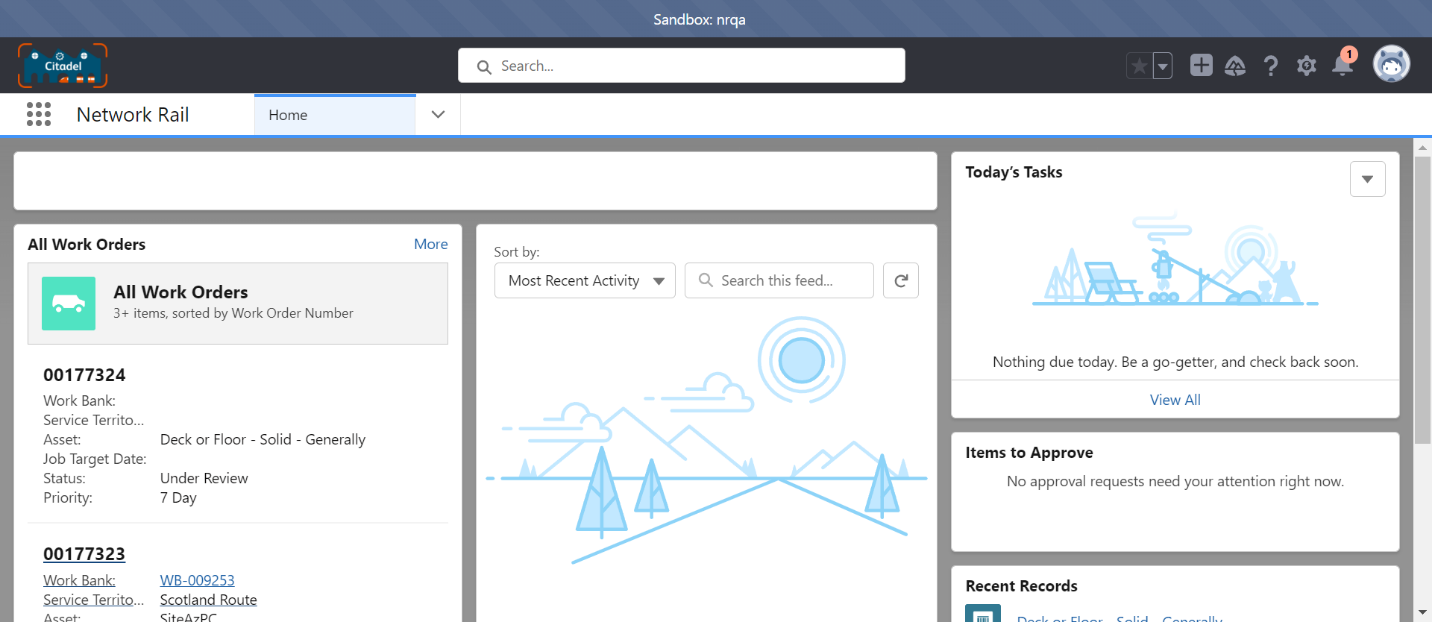
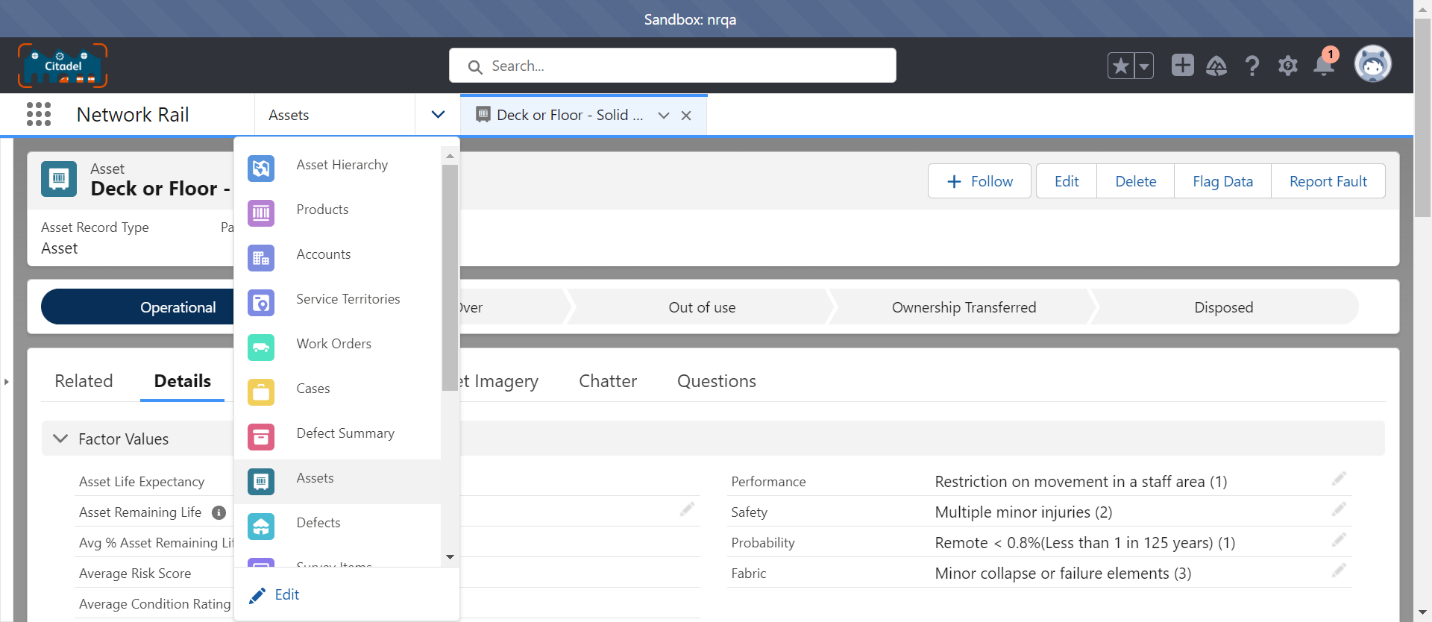
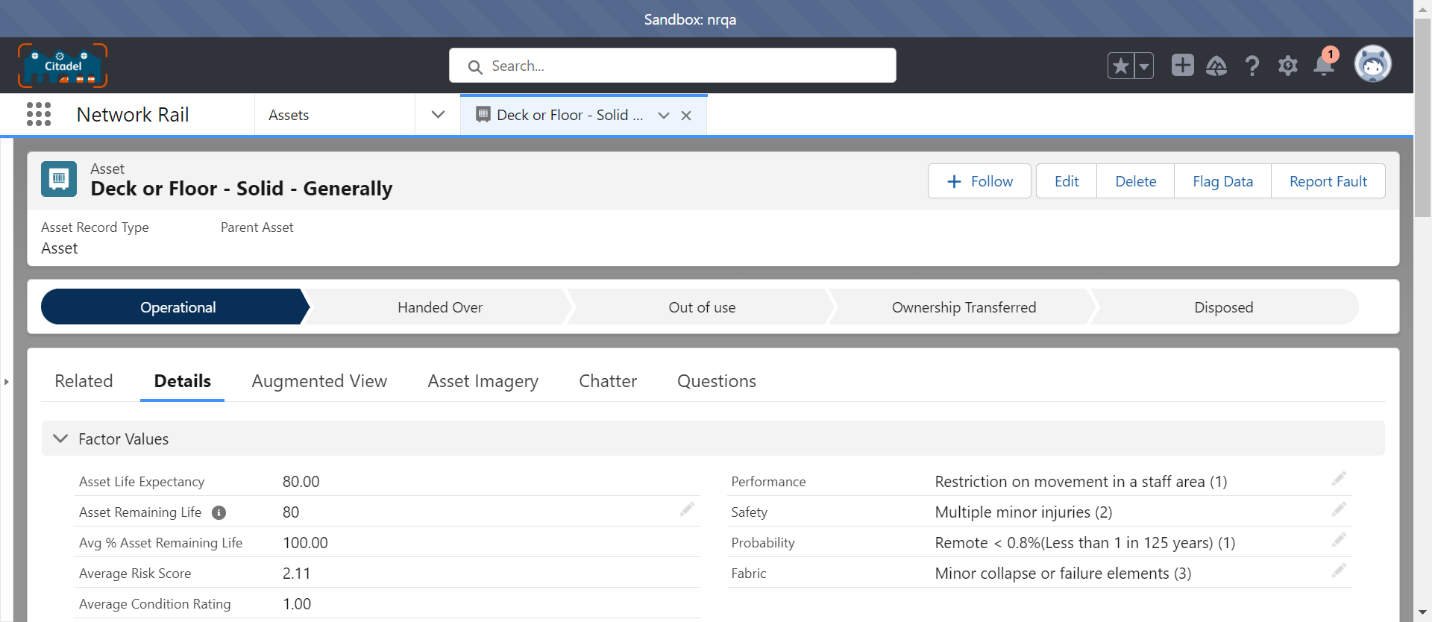
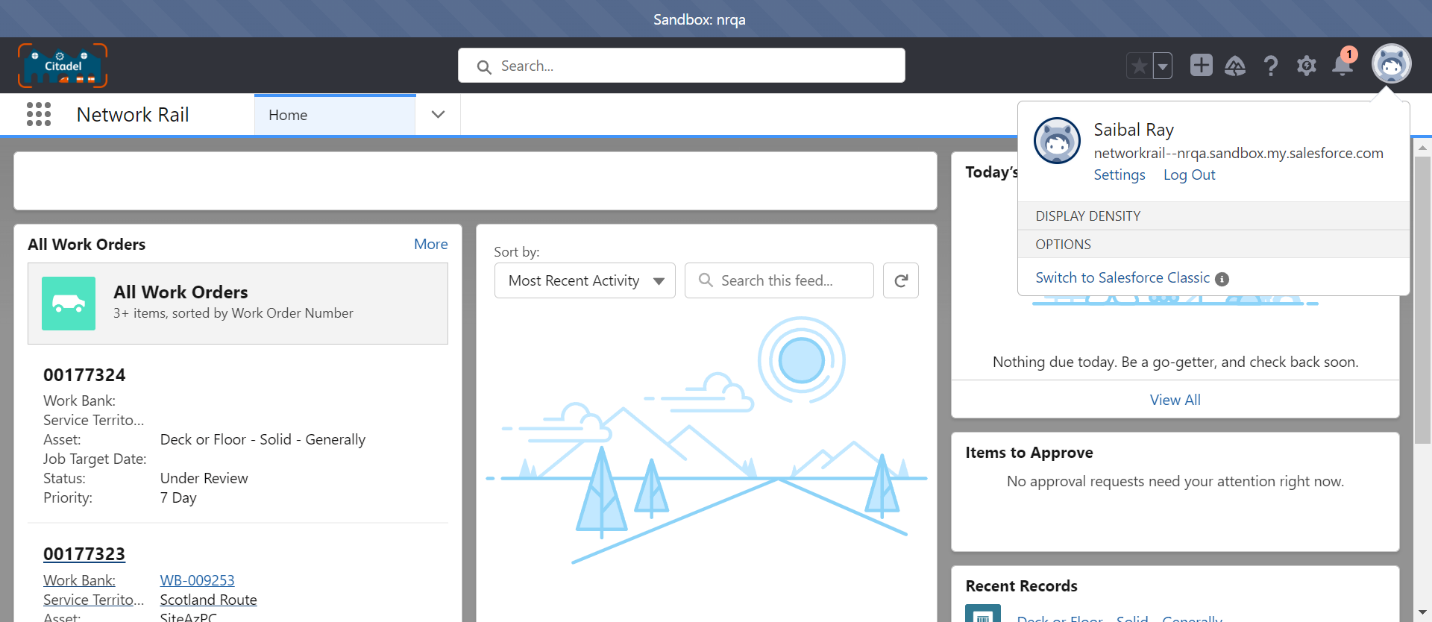
1. **Given**Salesforce Application Open
2. **When**Enter the Salesforce Application Credentials for <Role>
3. **Then**Salesforce Application Login verification
4. **Then**Verify global search box is displayed
5. **When**Salesforce Application Click dropdown tab and select Item <itemName>
6. **When**Salesforce Application Logout
7. **Then**Salesforce Application Logout verification

**Examples**:

| Role | itemName |
| --- | --- |
| HelpDesk Operator | Home |

@HomePage

**Scenario Outline**: Verify the fields which is present in-home page, app launcher, Global search bar, login & logout functionalities

1. **Given**Salesforce Application Open
2. **When**Enter the Salesforce Application Credentials for HelpDesk Operator
3. **Then**Salesforce Application Login verification
4. **Then**Verify global search box is displayed
5. **When**Salesforce Application Click dropdown tab and select Item Home
6. **When**Salesforce Application Logout
7. **Then**Salesforce Application Logout verification