1. Created Maven project and added testng and selenium dependencies in pom.xml.
2. Used TestNG framework to execute testcases.
3. Created DriverFactory.Java class to initialize the driver instance.

Based on the property, System.getProperty("browserType") driver will be initialized.

1. All ui object identifiers are maintained in Locators.java class.
2. Create Table.Java and TableRow.java classes to handle table operations.
3. Followed page object model pattern and created a class for each ui page.

For all the UI pages, BasePage.java is the super class. In this super class, driver variable is maintained, and it is available to all sub classes.

Also opening the browser and closing the browser is handled in this super class.

1. For menu navigation created abstract classes. IAccountMenu for account menus navigation, IEntityMenu for Entities menu navigation.

In future if any new menu is added, just extend this class.

Navigating to Entity and Account menu is handled in Application class.

public BasePage NavigateToAccountMenu(IAccountMenu accountMenu)

public BasePage NavigateToEntityMenu(IEntityMenu entityMenu)

Ex: Navigating to Entity -> Branch

new application().NavigateToAccountMenu(new BranchMenu()).

I applied dependencyInversion principle and open closed principle.

Also we can navigate to any menu from any ui page.

Ex: new BranchPage().NavigateToAccountMenu(new BranchMenu());

1. Created ICreateOrEditModal.java class to handle create/edit of branch and staff entities.
2. All validations are written in validators folder.
3. Data driven testing is achieved by generating dynamic object at runtime.

Ex: to test create branch

BranchInfo branch= DataUtility.*getRandomBranchInfo*();

CreateBranchWithValidData(branch);

**To Run Tetcases:**

Set the below properties in @beforeMethod

System.*setProperty*("browserType", "Chrome");

System.*setProperty*("DriverLocation", "C:\\Selenium\\chromedriver.exe");

I only implemented for chrome.

**Testcases Covered:**

1. Create/View/edit/Delete/search of Branch and staff.
2. Verified pagination of staff.
3. Password reset (verification not implemented as it is throwing error after password reset)
4. User settings updation(verification not implemented as it is throwing error)
5. Register new user ((verification not implemented as it is throwing error)
6. Login and logout