Task

- Use the framework to launch the https://magento.com/

- Navigate to the login page as shown below

- Click on Forgot password and then click on Submit button to validate the below error

- And also provide mail id as “bala#$$” and click on submit to check for below validation as

well.

Note: Two assertion can be done in single test method.

Test Code : In this code I have written 2 test method in Log in page for which we had already a framework was given. After adding those 2 method I have called them in main method.

Login Page :

```
package com.ibm.magentopages;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.How;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class LoginPage {

	@FindBy(id="email")
	WebElement emailEle;

	@FindBy(how=How.ID,using="pass")
	WebElement passEle;

	@FindBy(xpath="//button[@type='submit']")
```

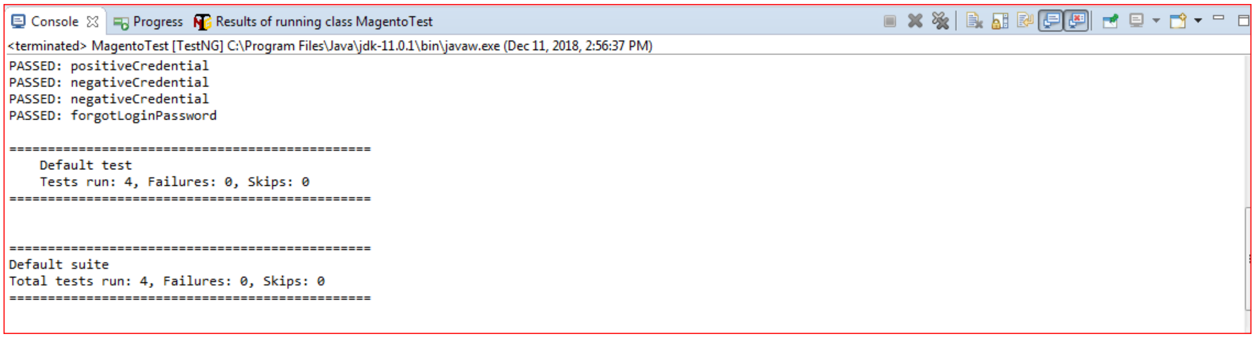
```
        WebElement loginEle;

        By errMsgLoc=By.className("error-msg");

        @FindBy(xpath="//a[@href=\"https://account.magento.com/customer/account/forgotpassword/\"]")
        WebElement forgotPassword;//  this is the locator to click on forgot password link


        @FindBy(xpath="//input[@id='email_address']")
        WebElement enterEmail;

        @FindBy(xpath="//span[text()='Submit']")
        WebElement submitButton;

        By requiredErrorMsg=By.className("validation-advice");

        By enterValidEmailAdd=By.className("validation-advice");

        WebDriverWait wait;
        WebDriver driver;

        public LoginPage(WebDriver driver, WebDriverWait wait) {
                PageFactory.initElements(driver, this);
                this.driver=driver;
                this.wait=wait;
        }

        public void enterEmailAddress(String userName)
        {
                emailEle.sendKeys(userName);
        }

        public void enterPassword(String password)
        {
                passEle.sendKeys(password);
        }

        public void clickOnLogin()
        {
                loginEle.click();
        }

        public String getPageSourceforInvalidErrorMessage()
        {
                wait.until(ExpectedConditions.presenceOfElementLocated(errMsgLoc));

                return driver.getPageSource();
        }

//  till above code  was there so I have used that only and have added the  two method as shown below.
```

```
	public void forgotPassword()  //  this method will click on  forgot password
link
	{
		forgotPassword.click();
	}

	public void enterEmail(String reenteremailaddress)
	{
		enterEmail.sendKeys(reenteremailaddress);//  this method to enter the
email address
	}
	public void submitButton()//  this method will click on the submit button
	{
		submitButton.click();
	}
	public String requiredFieldErrorMessage()//  this method is to validate the
page source and do the aasertion
	{

	wait.until(ExpectedConditions.presenceOfElementLocated(requiredErrorMsg));

		return driver.getPageSource();
	}
	public String getPageSourceToEnterValidEmailErrorMessage()
	{

	wait.until(ExpectedConditions.presenceOfElementLocated(enterValidEmailAdd));

		return driver.getPageSource();
	}
}
```

Main Method : Test case 1 and 2 were already given and we been asked to add one more test cases as per the requirement so I have added the test case 3

```
    @Test(priority=3, testName="Check Forgot Email validation")
      public void forgotLoginPassword() throws IOException
      {


             String url = data.get("url");
             String reenteremailaddress=data.get("reenteremailaddress");
             driver.get(url);

             HomePage home = new HomePage(driver);
             home.clickOnMyAccountIcon();

             LoginPage login = new LoginPage(driver, wait);
             login.forgotPassword();
             login.submitButton();

             String pageSourceOne=login.requiredFieldErrorMessage();

             Assert.assertTrue(pageSourceOne.contains("This is a required field."),
"Assertion on Required Field");//  this is assertion  that is verifying the when we
don’t enter anyemail address  and click submit

             login.enterEmail(reenteremailaddress);
             login.submitButton();

             String pageSourceTwo=login.getPageSourceToEnterValidEmailErrorMessage();

             Assert.assertTrue(pageSourceTwo.contains("Please enter a valid email
address."), "Assertion on Enter Valid Email Address");//  this assertion verifies
that we enter invalid emailaddress and click on submit.
      }


}
```

Screen Shot from console:



<terminated> MagentoTest [TestNG] C:\Program Files\Java\jdk-11.0.1\bin\javaw.exe (Dec 11, 2018, 2:56:37 PM)

```
PASSED: positiveCredential
PASSED: negativeCredential
PASSED: negativeCredential
PASSED: forgotLoginPassword

===============================================
    Default test
    Tests run: 4, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 4, Failures: 0, Skips: 0
===============================================
```