1. Write 10 test cases for this login page- https://portal.guardrfid.com/. You can use any template and submit this in word or excel format.
   The requirement statement for this login page is- A login page for end users should have the following fields- username, password, login button, forgot password. You can make assumptions related to validations on each of these fields.

   Test Case 1: Check the results when invalid username or password is entered.

   Test Case 2: Check the results when valid user credentials are entered.

   Test Case 3: Check the results when a field is left empty and login button is pressed.

   Test Case 4: Check if the user credentials are remembered for the next login when user checks the remember me checkbox.

   Test Case 5: Check user can login with new password and not with old password, upon resetting the password.

   Test Case 6: Verify the login against SQL injections.

   Test Case 7: Check the timeout feature of the login session.

   Test Case 8: Check the characters entered cannot exceed the limit specified for length of username and password.

   Test Case 9: Check if it allows to login with different or same user credentials on different tabs at same time.

   Test Case 10: Check the compatibility of the design on devices with different screen sizes.

2. Write a program in java to check if a given String is a palindrome or not. Use Junit to write unit test cases for this program.

```java
public class Palindrome {

    public static boolean isEqual(char a, char b){
        if( a == b)
            return true;
        return false;
    }

    public boolean isPalindrome(String input){
        //Converting input to Lowercase and removing all white spaces
        input = input.toLowerCase().replaceAll("\\s+", "");

        for(int i =0, j= input.length() - 1; i < j; i++, j--)
            if(!isEqual(input.charAt(i),input.charAt(j)))
                return false;
        return true;
    }
    Run | Debug
    public static void main(String[] args) {
        Palindrome palindrome = new Palindrome();
        System.out.println(palindrome.isPalindrome("HEllo"));
        System.out.println(palindrome.isPalindrome("NON"));
        System.out.println(palindrome.isPalindrome("MIISSIIM"));
    }

}
```

Figure 1: Implementation of method to check if a string is a palindrome or not

```java
Run Test | Debug Test | ✓
public class TestPalindrome{
    private String input;
    private Palindrome palindrome;

    @Before
    public void setUp() throws Exception {

        input = null;
        palindrome = new Palindrome();

    }

    @Test
    Run Test | Debug Test | ✓
    public void emptyStringPalindromeTest(){
        input = "";
        assertTrue(palindrome.isPalindrome(input));
    }

    @Test
    Run Test | Debug Test | ✓
    public void singleCharTest(){
        input = "A";
        assertTrue(palindrome.isPalindrome(input));
    }

    @Test
    Run Test | Debug Test | ✓
    public void twoCharPalindromeTest(){
        input = "AA";
        assertTrue(palindrome.isPalindrome(input));
    }

    @Test
    Run Test | Debug Test | ✓
    public void twoCharNotPalindromeTest(){
        input = "AS";
        assertFalse(palindrome.isPalindrome(input));

    }

    @Test
    Run Test | Debug Test | ✓
    public void threeCharPalindromeTest(){
        input = "ASA";
        assertTrue(palindrome.isPalindrome(input));
    }
```

Figure 2:Junit test cases

```java
    @Test
    Run Test | Debug Test | ✓
    public void threeCharNotPalindromeTest(){
        input = "ASD";
        assertFalse(palindrome.isPalindrome(input));
    }

    @Test
    Run Test | Debug Test | ✓
    public void evenLengthPalindromeTest(){
        input = "ASDFFDSA";
        assertTrue(palindrome.isPalindrome(input));
    }

    @Test
    Run Test | Debug Test | ✓
    public void evenLengthNotPalindromeTest(){
        input = "ASDFGDSA";
        assertFalse(palindrome.isPalindrome(input));
    }

    @Test
    Run Test | Debug Test | ✓
    public void oddLengthPalindromeTest(){
        input = "ASDFDSA";
        assertTrue(palindrome.isPalindrome(input));
    }

    @Test
    Run Test | Debug Test | ✓
    public void oddLengthNotPalindromeTest(){
        input = "ASDFDSP";
        assertFalse(palindrome.isPalindrome(input));
    }

    @Test
    Run Test | Debug Test | ✓
    public void whiteSpacePalindromeTest() {
        input = "AS DFGH  JK   JHGF DSA";
        assertTrue(palindrome.isPalindrome(input));
    }

    @Test
    Run Test | Debug Test | ✓
    public void alphaNumericPalindromeTest() {
        input = "AS 121 SA";
        assertTrue(palindrome.isPalindrome(input));
    }
}
```

Figure 3:Junit test cases cont.

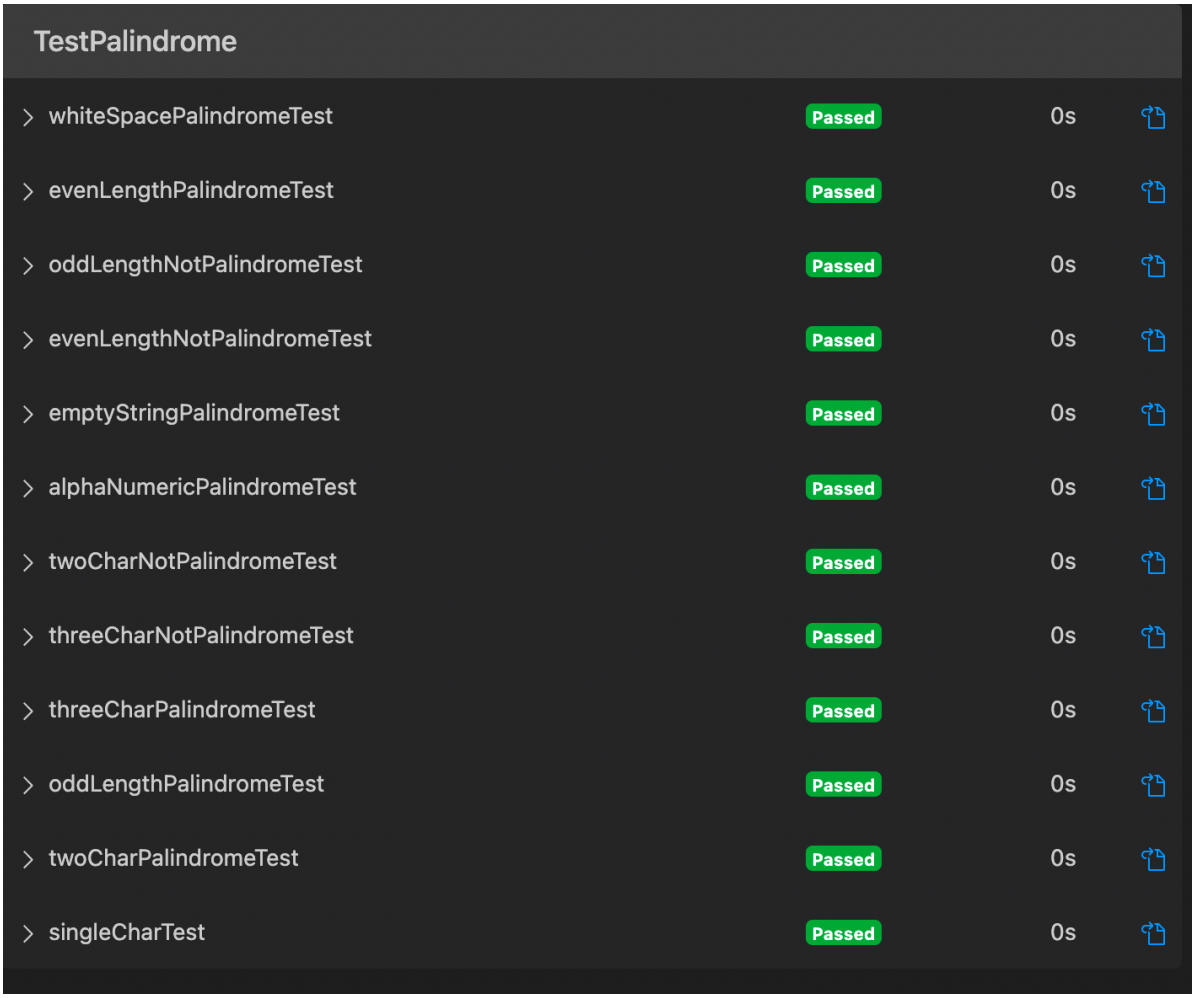

Figure 4: Junit test results