Manual Testing : Feature to be tested 'Forgot Password'

Goals:

1. Ensure that user can retrieve the password link in their email and able to reset password
2. Ensure that hacker or bad actor do not abuse the system by sending consecutive requests in short span (Avoiding DDoS)
3. Verify that hacker can't guess the usernames and there by password
4. Verify that User can login to shipt website with newly created password after receiving an reset password link via email

Pre-Conditions:

1. Ensure that www.shipt .com is accessible
2. Close popup window, if any
3. Click on Login Button on Home Page
4. Click on Forgot Password Link on Login Page
5. Preregister to shipt with access to email

1. Navigate to www.shipt.com and describe the testcase

Test Case/Step(s):

1. Verify Forgot Password Feature Page (FPFP) contains all text, fonts or Page title
2. Verify Textbox and Button are active / enabled
3. Verify FPFP does not send reset password emails with invalid emails like "aa"
4. Verify FPFP does show Java Script with <script>alert('aaa')</script> to avoid SQL Injection
5. Verify FPFP does not send emails with invalid emails like aa@aa.com
6. Verify FPFP does not send emails in succession (To avoid DDoS)
7. Verify FPFP does not show meaningful warning thus revealing users' table contents (Always show a generic message like 'An email has been sent'. However, the system should actually send email if it is legitimate email, otherwise throw error in the server log file)
8. Verify FPFP sends email with valid email
9. Verify that Reset Password link expires after resetting the password
10. Verify that Reset Password link expires when user gets another/new Reset Password link
11. Verify that Database Table(s) track which user asked to resend the link and when
12. Verify that Server log shows appropriate errors when user enters wrong/invalid emails or valid emails

Note: It is assumed that QA Engineer properly identifies Requirement/Specification Number vs Test Case Numbers so that QA Lead can create Requirement Traceability Matrix Document for Product Managers/Stake Holders to review.

Note: Not all the above testcases are automated. Only the major ones (or feasible) are automated to demonstrate the proficiency in Automation. Also, Automation testing is not 100% replacement for manual testing.

2. Locate one Bug

I found this bug less than one minute and hence pursued further using exploratory testing

- Behavior/Error: I see that Forgot Password Page shows if the email exists in the Users Table. It states that 'aa@aa.com is not a valid email".
- Suggestion to Correct: Just print "A reset email link has been sent to aa@aa.com" . This gives no idea to hacker whether this email exist or not in the DB.

3. Priority of Bug

- Priority will be set by PMs not QA Engineers. However, I would suggest P1 not because this is really P1. But it is easy to fix. Low cost and high ROI.
- Now a days most of hackers are hacking consumer facing websites and any data breach would be devastating to companies and ultimately lead to bankruptcy.
- Finally, I would present my suggestion with PM or upper management and take appropriate decision in Bug triage meeting

Automation

- The automating framework is developed as Page Object Model using Java/TestNG/Maven/Selenium/Log4j/JavaScript etc
- The reason I chose Java instead of Ruby is that there is lot of community support as compared to Ruby. Also, third party integration tools are available for Java or Python
- Also, for API testing I chose RestAssured framework which works with Java
- For locating the elements, I use custom css locators when id, name are not available. I take help from Chrome Developer Tool.
- Common causes of instability: Automating still-in-Dev, not well designed framework (eg: not using POM frame work etc), trying to automate complex tasks like verifying server logs etc.
- I recommend automating test cases that are stable apps using well proven framework, technologies so that tests will be consistent and easy to debug.
- /src/main/java and /src/test/java have source code.

How to run: There is a testing file in src/test/runners folder. Right click and run as testng file src/test/runners/testng_ui_and_api_test_suite.xml. On my end, I used Jenkins which is popular CI tool to run automatically.

API Testing

- The automating framework is developed using Java/TestNG/Maven/RestAssured/Log4j/Groovy etc
- /src/main/java and /src/test/java have source code.
-

Please note that I use postman for manually testing the APIs. Here, I tried to run them using existing automation framework using Jenkins CI as it is easier/stable/easy maintainable to write API tests.

- How to run: There is a testing file in src/test/runners folder. Right click and run as testng file src/test/runners/testng_ui_and_api_test_suite.xml. On my end, I used Jenkins which is popular CI tool to run automatically.

Note: Log file is generated under ../shipt/log4j_output.html
Note: Test Reports are generated under ../shipt/test-output/emailable-report.html

SQL

Note: I tested these queries using postgres 9

1. List me the stores allowed to sell alcohol

```
select name from interview.stores where allowed_alcohol is true    -- gives as Gettar
```

2. Give the product name of the 2 most expensive items based on their price at store id 1

```
select p.name from interview.products p
join interview.store_prices s on p.id = s.product_id
where s.store_id = 1 order by price desc limit 2;      --gives Golden Banana and banana
```

3. List product that are not sold in Store id 2

```
select p.name from interview.products p
join interview.store_prices s on p.id = s.product_id
where s.store_id = 1
except
select p.name from interview.products p
join interview.store_prices s on p.id = s.product_id
where s.store_id = 2;  --gives Golden Banana and banana
```

4. Most popular product Sold

```
select count(*), p.name from interview.products p
```

```
join interview.store_prices s on p.id = s.product_id
group by p.name having count(*) > 1  -- gives Grapes
```

5. Update the line_totla field
   - I updated just one field and verified if it is indeed verified

```
update interview.order_lines set line_total = 5 where id = 1
select * from interview.order_lines
```