Upgrade QA Challenge Test cases

WEB UI:

Scenario 1:

**BasicInfoCojoinTest** : This test covers the positive flow for inputting Basic info for both main and co-joint applicant

Output: This validates if after filling up info it gets to Income page successfully

Scenario 2:

**BasicInfoIndividualTest** : This test covers the positive flow for inputting Basic info for only individual applicant

Output: This validates if after filling up info it gets to Income page successfully

Scenario 3:

**CheckRatepageInvalidLoanAmount** : This test will validate inputting invalid Loan amount with All valid Loan purpose

Output: for all test data users should get an error message

Scenario 4:

**CheckRatepagevalidLoanAmount** : This test will validate inputting valid Loan amount with All valid Loan purpose

Output: for all test data user should be able to get to Basic info page successfully

Scenario 5:

**EndToEndAccountValidationTest**: This test will validate END to END task given in QA challenge.

Output: Once the user logs back in, user should be able to see same Loan amount, APR, LOAN term and Monthly payment as he entered previously

**Other Scenarios not covered in WEB Automation:**

**Invalid Inputs**: All pages like rate page, offer page, income page etc can be validated on each and every input fields like date, age, valid state , city, username etc. but this can be easily covered in the solution provided .

**Login Scenarios**: 1. Once the user logs in, copy the URL ->sign out -> go to login page and try to log in with previous Copied URL

Output: user should go back to log in page

2. Login in to 2 different Tabs, log out from 1 TAB and refresh another TAB or try to access some contents

Output: user should log out from system

3. Login to two different browsers at same time.

Output: user should be able to log in successfully

4. Log out from 1 of the browsers and try to access account in another Browser. User should log out from that browser as well (but it depends on the if we have the feature or Not)

5. Deactivate/block the user in the system, when user try logging back in, he should not be able to log back in.

6. Make sure user cannot log in with putting a SQL query in username or in URL (SQL injection) and NO sensitive info should be shown in ERROR

7 make sure multiple attempts are blocked with wrong username or password

8. make sure user can change his password and can log back in with new password

9. Make sure if user put wrong password, then Error message should not be obvious if it was username was wrong or password was wrong.

10 Make sure Password policies are followed for creating an account

11. check Maximum length for username and password policy

12 Make sure when session expires the user should not be able to access any contents and log out.

13 make sure we are not passing info if user click on any Pop-up like fraud ads etc.

14 Make sure username and passwords are not passed in URL or in HTML DOM

15 Make sure user cannot pass any validation on by using back-end API call. So, all validation should be validated both sides.

16. make sure captcha feature is available, if someone trying to login multiple times

17 Make sure once the user Logout, back button on the browser should not log user back in

18. Make sure same user can log in two different Tabs at same time . Same applies to two different browser.

API Test cases:

Scenario 1 : -

**upgadeAPIWithvalidLoanAppid** : User sends a request with valid Loanappid and all valid parameters

output: user should be able to receive a valid response back with 200 OK status with valid length for Loanappid in response

Scenario 2:

**upgadeAPIWithInvalidLoanAppid :** User sends a request with Invalid Loanappid and all valid parameters

output: user should get a 404 NOT found Error

Scenario 3:

**UpgradeApiWithInvalidMethodTest** : User sends a request with valid Loanappid and all valid parameters BUT with GET method

Output : user get 405 Error in response

Scenario 4:

**UpgradeApiWithInvalidheadersTest:** User sends a request with valid Loanappid , but wrong headers

Output: user gets 500 status code in the response

Scenario 5 :

**UpgradeApiWithNoPayLoadTest :** User sends a request with no payload

Output: user gets 500 status code in the response