#AutomATAhon2024
23rd November 2024

#AutomATAhon2024 is brought to you by CPSAT

https://cpsat.agiletestingalliance.org/

CP-SAT Foundation

This is the foundation level exam to assess your Selenium Eco System knowledge using Multiple choice questions

CP-SAT Advanced

This is the Advanced level exam to assess your Selenium Eco System knowledge using Practical Scenario based questions

CP-SAT Expert

This is an expert level exam to assess how you can design an end to end, cross browser scalable, continuous testing automation solution

Selenium Certificate Exam

Table of Contents

AutomATAhon2024 Challenge – Round 1

Introduction

#AutomATAhon2024 is 5th edition of the automation challenge is brought to by CPSAT and is part of #ATAGTR2024

Details for the hackathon are available on the below page

https://gtr.agiletestingalliance.org/automatahon/

Rules

Please read the below rules before starting the challenge.

1. It is a time boxed event – which means a fixed duration will be provided to all the participants to solve the challenge. You are given total 4 hours starting 1 PM IST to complete the challenge. All servers would stop at 5 PM IST. You have to ensure that the codebase used by you is checkedin in your own private GIT repository before 5 PM. The evaluation will happen from the codebase that you have checked in before 5 pm IST. Any updates after 5 PM IST will not be accepted.
2. You can use any ~~open source~~ automation tool(s). You do not need to setup the tool or framework on Jury machines.
3. Round 1 would be held over Zoom/MS Teams (Virtual) and Round 2 will be conducted live during Conference location in the form of a live Quiz. Final Rankings will be based on 90% weightage from Round 1 and 10% weightage from Round 2.
4. It is a team event. Maximum members in a team allowed are two (2)
5. One laptop per team member is allowed and each team member should sign in from the same laptop that is used for coding/automation to the Zoom/MS Teams meeting
6. The competition will be hosted using Microsoft Teams or Zoom, you need to ensure that you have access to Microsoft Teams / Zoom from your laptop and you need to turn video camera showing both the team members during the entire duration of the event. One team member has to share the screen showing what is being done. You should have stable internet connection during the competition so that video sharing is not hampered by bandwidth. If the video is not shared, it may be a reason for disqualification.
7. Both the team members will be allocated a virtual room in the Zoom meeting, they must be present in that virtual room during the entire duration of competition. They can be physically present in different locations. Both the team members must keep their cameras on in the virtual room. One team member has to share their laptop / desktop screen. Jury members can ask other member also to share his/her screens on demand. One screen sharing (from one of the participants have to be continuously shared)

8. There is no cap on the number of teams/registrations from a particular organization
9. Use cases on API Testing, Mobile, UI and Functional Testing on multiple browsers, Reporting and other use cases may be there.
10. Problem statements will be provided to all the teams at the start of the event. Participants would have to use their own private GIT Repo. Participants must automate the problem statement and hook with a CI system like Jenkins.
11. Participants must use their own Jenkins or similar CI tool setup. They can have the setup ready before the competition starts.
12. **Team has to move the code into the GIT repo as per the scheduled time**. Last checkin time to the GIT repo would be considered as the final submission time. Participants may have to make ATA Jury member as a collaborator in case required.
13. Participants need to have a good internet bandwidth during the competition.
14. **It is advisable to use personal laptops rather than company, the application used for automation challenge may be hosted on IP address which may not be accessible from company's VPN or company provided laptop.**
15. **The server on which the application is hosted will be turned off after the time limit is reached, please ensure that the code checkin to git repository and execution readiness is achieved during the allocated time.**
16. During the Demo section of the competition, each team has to execute their code from the GIT code base. A new test environment (new server) may be given for the test execution, it is recommended that the pre and post test execution preparation should be planned for your tests to run seamlessly.
17. Evaluation would be done based on completion of tasks/challenges. On completion and successful execution of challenge you score full marks associated with that challenge and **NO marks will be given if the challenge is not completed. Partial completion of tasks will get NO marks.**

    In case of a tie-on total marks both in the round 1 and round 2, the tie breaker will be based on two parameters, time taken to solve all the challenges and best practices followed

    Speed of solving the challenge (time stamp of the final code push in the repository) will be used as the first tie breaker in case the marks are equal and then on best practices.

    some of the best practices parameters are listed below.

    - Quality and stability of automation
    - Ability to modify tests (changes, addition and deletion of test scenarios)
    - Modularity
    - Reusability

- Scalability
- Adaptability
- Design methodology
- Code quality
- Flexibility to integrate with CI tools

18. **Jury members will not be evaluating the team members from their own organization**.
19. Jury will select the top **5 entries or more** for the final representation on 8th of December 2024
20. **Non-participation of the teams during the conference (8th of December 2024) would be considered a reason for disqualification. All teams selected for final round hence must be present during the conference.**
21. **Jury's decision would be final.**

Challenge

Part 1 : 1 PM IST to 5 PM IST

Duration 4 Hours – for understand challenges, coding, execution and testing, checkin code to Git repository

Part 2 : 5:15 PM to 8:15 PM

Demo to Jury from 5 PM onwards. Each team will be allocated 15 minutes each to showcase their execution and any QnA from Juries on the chosen CI platform using the private GIT repo.

Important Challenge Rules

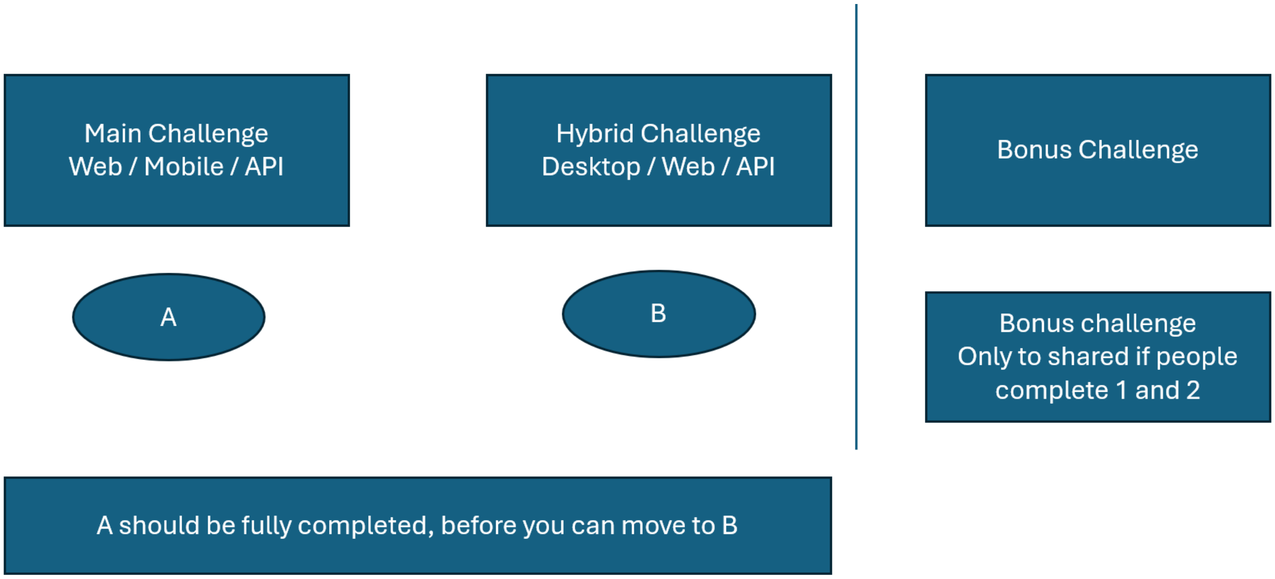
- Evaluation would be done based on completion of tasks/challenges. On completion and successful execution of challenge you score full marks associated with that challenge and **NO marks will be given if the challenge tasks are not completed fully**. **Partial completion of tasks will get NO marks.**
- **You have to share your private git repository url's (that you are going to use). All Challenges have to executed in a CI environment from your git repositories.**

- **You are supposed to maintain the private repositories till end of #ATAGTR204 on 8th December. Jury can review the repository during the conference. It is required to bring the same laptop to conference as you used for the competition. You may be required to demonstrate the automation in case needed on 8th of December.**

- **Any out of sequence tasks will not be considered for evaluation.**

About Challenges

There are three challenges in all. 3rd Challenge is a bonus challenge which would be shared only if the first two challenges are completed fully. You may ask Jury for the same.



Main Challenge
Web / Mobile / API

Hybrid Challenge
Desktop / Web / API

Bonus Challenge

A

B

Bonus challenge
Only to shared if people complete 1 and 2

A should be fully completed, before you can move to B

Challenge A has total 13 tasks and Challenge B has a total of 3 tasks

It is important that teams follow the sequence. A.1 to A.13 and then move to Challenge B.1 to B.3

A.1 should be completed before A.2 can be completed and so forth

Any out of sequence tasks will not be considered for markings.

If you have completed till A.3 , the next to be evaluated would be A.4. Any other tasks beyond A.4 would not be considered done.

AutomATAhon2024 Challenge - Round 1

Summary Mark breakdown for Challenge A and Challenge B

Below table summarises the marks breakdown of the two challenges.

Challenge A		265
A.1	**Setting up Library in Koha (UI)**	5
A.2	**Creation of librarian under Library created in Step #1**	20
	API part	
	UI Part	
A.3	Creation of patrons for OPAC	30
	○ OPAC UI	
	○ Librarian UI	
	○ API	
A.4	**Validate that all the patron created in Challenge #3 are allowed to login**	10
A.5 AB	Multi browser parallel challenge	40
	A) Creation of Book & Items under Cataloguing : do it in (Browser 1)	
	B) Validate book is available in OPAC UI Interface (do it in (Browser 2)	
A.6 AB	Multi browser parallel challenge	40
	A) Add items to Book (Staff UI Interface)	
	B) Switch to OPAC UI & validate the reflection of its availability	
A.7	Add the book to your card and place hold	10
A.8 AB	Multi browser parallel challenge	30
	A. Open patron & checkout book item from Staff UI interface	
	B) Switch to OPAC UI Interface and validate that the book is now issued to patron	
A.9 AB	9 A is API and 9 B is Multi browser	20
	A) Using Koha API, add credits to the currently logged in patron	
	B) Switch to OPAC UI / Staff Interface & validate that the credit amount is reflected for the patron	
A.10	Submit or check in the book in library from Staff UI interface	10
A.11	Switch to OPAC UI & validate that the book is no longer issued to patron	10
A.12	On OPAC UI, search for the book and validate holdings availability	10
A.13	Book cover image comparison in OPAC UI	30

Challenge B		100
B.1	Desktop app automation – 50 marks	50
B.2	(Web app automation) – 30 Marks	30
B.3	Koha API – 20 Marks	20

AutomATAhon2024 Challenge - Round 1

Challenge A and Challenge B – both use KOHA

About KOHA

About Koha: *Koha is a free, open-source integrated library system (ILS) that helps libraries manage their collections, automate workflows, and improve user experiences*

Koha Includes module for following:

- **Online Public Access Catalogue (OPAC)**: A simple interface for users to search for and reserve items, suggest new items, and more
- **Full catalogue**: Allows library staff to capture details of all library items
- **Circulation**: Automates borrowing and item management, and integrates with the OPAC
- **Acquisitions**: Helps librarians with acquisitions and budget management
- **Serials management**: Performs functions related to serials
- **Reporting**: Performs functions related to reporting

Note: Before you begin with the challenges, please take your time to understand the Koha by using this link (LINK TO CHAT GPT THREAD). We have created a ChatGPT thread for you to easily understand few key keywords and/or jargons of Koha that will help you familiarize yourself with. Feel free to use your googling skill to get deeper understanding of Koha.

Here is the link to official documentation for Koha

Koha User Manual	Koha Manual (en) — Koha Manual documentation
Koha API Documentation	Koha REST API

Staff UI/Librarian UI Interface : http://34.148.101.249:8081/

OPAC / Patron UI : http://34.148.101.249:8080/

Koha Staff Interface Admin Credentials (Please ask from the Jury members)

AutomATAhon2024 Challenge - Round 1

Please use your team admin credentials only. If we see any cross usage of admin credentials, the teams stands to get disqualified.

Team	Username	Password
Team 1 – The Sustainers		
Team 2 – Bug Busters		
Team 3 – AutoMinds		
Team 4 – AutoNinjas		
Team 5 – Test Titans		
Team 6 – AutoSurgeons		
Team 7 – GlitchHunters		
Team 8 – The Autobots		
Team 9 – AutoWarriors		
Team 10 – Ninja Coders		
Team 11 – Test Verse		

Challenge A and Challenge B are in next sections.

Pls read next pages

Challenge A

1. **Setting up Library in Koha (UI)** – 5 Marks

   a. Open Staff interface (IP address) and login using admin credentials
   b. Switch to Admin & add Library

   Rules:

   - Test should be data driven. i.e. The test data should be stored in a file. i.e. text, json or excel.
   - The pattern for library code should be "GTR_HACK_INITIALSOFTHELIBRARYNAME". *i.e. if the name of Library is "The Mother's Library" then the library code should be "GTR_HACK_TML"*. This should not be hardcoded; it should be generated based on the name of Library
   - The generated library code should be stored back in the same file for later use.

2. **Creation of librarian under Library created in Step #1 – 20 Marks**
   a. API Steps
      i. Add a patron of type "Staff". Patron should be assigned to the library you just created
      ii. Once added, create a password for the patron.
   b. UI Steps (Staff UI)
      i. Open Staff interface, search for the patron
      ii. Assert that the patron is visible in table
      iii. On Patron details page, validate the patron details that was added through API
      iv. Set permission to the patron from Patron > Paton Name > Details > More > Set Permission. Set "superlibrarian" as permission.
      v. Once saved, validate that the changes are save successfully
      vi. Logout of the current admin and create scenarios that validates positive & negative scenarios for the librarian to which you just have given access to.

   Rules

   - This test should be data driven. The patron details should be fetched from a file of your choice. The username generated by /patrons endpoint should be stored back into the same file and in the same row or the object.
   - Assertions are mandatory. Use appropriate assertion library for this

3. **Creation of patrons for OPAC – 30 Marks**

The patrons for OPAC can be created by 3 ways. i.e OPAC UI, By Librarian & API.
Patron to be created is of Type Student

- For this test, create an excel file with first column as "**Creation Mode**" whose value should be either of following
  - OPAC UI
  - Librarian UI
  - API
- The file should at **least have 9 records. 3 records for each "Creation Mode"**
- The test script should start reading file from first row and based on the **"Creation Mode"** for the row being used presently, it should make use of **respective mode** & create a patron of type "Student".
- The test data to be used for patron creation should be generated randomly and **once created, those details should be stored back in the same file against the current "Creation Mode".**
- The pattern for "Card Number" for each patron should be "LIBRARYNAMEINITIALS_YYYYMMDD_HHMMSS". i.e. If the library name is "The Mother's Library" the patron library card number should be "*TML _20241123130000*"
- The Home library for the patron should be the one that you created in Step #1
- For creation mode of OPAC UI (Self registration form has to be used for creation of student)
- For creation mode of Librarian UI (Staff UI) – use the librarian credentials created earlier for login and then create new Patron (Type Student)
- For "Creation Mode" as "API", use the relevant API for creation of Patron (type student)

  following validations should be carried out for API Creation
  - Status code
  - Response time
  - Request Body -> Schema Validation
  - Response Body -> Schema Validation
  - Response Body -> Data verification

  All new students created should be saved in the same test data file from where the creation mode was read.

4. **Validate that all the patron created in Challenge #3 are allowed to login – 10 Marks**

- This test should be data driven. Means the file in which the patron details are stored should be used **(earlier A.3 scenario).**
- For each record, you should open the OPAC interface, enter username & password.
- Screenshot should be captured before login to the system and should be stored in report of your choice as well as in code repository with file name pattern as

“OPAC_LOGIN_USERNAME_YYYYMMDD_HHMMSS.png”. i.e OPAC_LOGIN_ADITYA.GARG_20241123_131214.png”

- Once logged in verify that the user is logged in successfully. Capture the screenshot and store it in code repository and report. The pattern for the file should be “OPAC_LOGGED_IN_USERNAME_YYYYMMDD_HHMMSS.png”
- Once logged in, also validate the patron details for each patron by navigating to “Personal Details” section in OPAC. **The expected data should be picked up from the file that was created in Challenge #A.3.**

For the next tasks around challenge A around Koha, test automation of them should follow the below guidelines.

- You should open two browser instances. One will be used to login to “Staff UI Interface” & the other should be used to login for a random patron using “OPAC UI Interface”. (Staff UI work to be done in one browser, OPAC UI work in another browser)
- You should use the Librarian credentials created in Challenge #1 for login in “Staff UI Interface”
- The test should be built in a way that it should pick a random patron from the patrons that you have created in Challenge #3 and use it to login to “OPAC UI Interface”
- All the challenges should run as a single suite. And both the browser instance should always be up and open.
- You should manage both the driver instances and browser instances parallelly.
- The challenges are built in such a way that you will be required to do few UI steps within one interface which needs to be validated on the other interface.
- You should **NOT** use same credentials for both UI interfaces.

5 A) and 5 B) have to be done together. Total marks for 5 a) and 5 b) are – 40 Marks

5. A) Creation of Book & Items under Cataloguing : do it in (Browser 1)

a. Login to staff interface using your super librarian credentials (A.2)

b. Navigate to Cataloging > 🔍 New from Z39.50/SRU

c. From the pop up, search for any random book by its title, ISBN etc.

  i. You can use LC Catalog - Browse to find out sample book details which can be used to search in Z39.50/SRU

d. From the results, pick the random book and do following

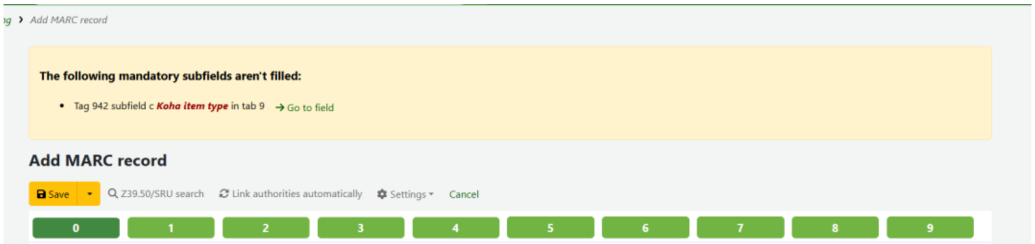
  i. Capture server name

  ii. Capture title name

  iii. From “Actions” column, click on “carat” and select “Import”

  iv. On Add MARC record screen, chose “Settings” and set the framework to “Books, Booklets, Workbooks”

  v. For “Save” button, select and option “Save and view record”. At this instance the book details should either be saved and you should navigate to the details page or you should see following error.



The following mandatory subfields aren't filled:

- Tag 942 subfield c **Koha item type** in tab 9 → Go to field

Add MARC record

At this instance, click on “Go to field” and select “Books” for the “Koha Item Type” mandatory field. Once selected click on “Save and view record”

vi. Once saved on book details page, validate the book title that you captured earlier

5. B) Validate book is available in OPAC UI Interface **(do it in (Browser 2)**

    e. Login to OPAC ui interface using any random patron credentials created by you.
    f. Use the title of the book as search string and search the book
    g. Validate that the book details are matching.
    h. Switch to Staff interface, capture the text visible against the field “Contributor(s)” and validate that they are the same in OPAC UI for the searched book.
    i. Capture the screenshot and store it in your report and code repository. The pattern for the file name should be ‘book-name-space-separated-by-hyphens-and-in-lowercase-opac-YYYYMMDD-HHMMSS.png’. i.e. if the book name is “Justine Ward and Solesmes” then the file name should be “*justine-ward-and-solemes-opac-20241123-131514.png*”

6 A) and 6 B) are also parallel browser challenges. 40 marks

6. A) Add items to Book (Staff UI Interface)

a. Switch to Staff UI interface, on book details page, click on “Edit > Manage Items”.
b. On Items screens, fill out values for following fields
    i. 2 - Source of classification or shelving scheme (Random value from dropdown)
    ii. Capture the text values for fields (Home Library, Current Library)
    iii. C. Shelving location as “Book cart”
    iv. D – Date acquired as any random date from past 3 months
    v. P – Barcode should have pattern “INITIALSOFLIBRARYNAME_BK_1”. This initial should be the initials of library you have created. i.e. if the library name is “The Mother’s Library” then Barcode should be “TML_BK_1”
    vi. Validate that the value at field “ y – Koha Item Type” is “Books”
    vii. Click on “Add multiple copies of this item” button available at the bottom of the page.
    viii. At field “Number of copies of this item to add”, generate any random number between 5 to 15, enter the generated number and click on “Add” button
    ix. Once added, at the items should be visible on top of the table.

x. Extract values from "Barcode" column and validate that it has the same barcode pattern as you added before and all subsequent record should have values in increasing order.

xi. Store all the barcodes in a file

6. **B) Switch to OPAC UI & validate the reflection of its availability**
   a. Switch to OPAC and search for the same book or if currently on the book details page then
      i. Validate that "Holdings" tab should have count which is equal to the number of items added in Challenge #6 A > viii
      ii. Validate that the for each row under "Holdings" tab, the value of "Current Library" is same as what was captured in Challenge #6 A > ii
      iii. Validate that the "Status" for all the rows should be "Available"

7. **Add the book to your card and place hold – 10 Marks**
   a. Switch to OPAC UI interface, search for book and navigate to book details page if not already at details page
   b. Click on "Add to your cart"
   c. Once added, validate that text next that previously read "Add to your cart" should now read "In your cart (remove)"
   d. Validate that the number against the "Cart" icon available on top header next to Koha logo should be equals to 1
   e. Click on "Cart" icon, select the checkbox against the book and click on "Place Hold"
   f. On "Placing a hold" screen, select the pickup location as the library name captured in Challenge #6 A > ii
   g. Click on "Confirm hold"
   h. Once confirmed, validate that the items/book details are visible in "Holds" tab for the current OPAC user.
   i. Validate details such as
      i. Book title > should be one that you added
      ii. Place on > should be Today's date
      iii. Status > should be "Pending"

8 A and 8 B below have to be executed in parallel browsers, together. Total marks = 30

8. **A) Open patron & checkout book item from Staff UI interface**
   a. Switch to Staff UI interface, search for the patron with whom the book was placed on hold and open patron details page
   b. At the left navigation go to "Holds History"
   c. Validate that the book that was placed on hold should be visible under "Holds History" table.
   d. Validate that the status of the book should be "Pending"
   e. At left navigation, click on "Check out" tab

f. Navigate to "Holds" tab on Checkouts page and validate that the book details are matching along with hold date

g. On checkouts page, at checking out to section, enter any random barcode that was generated and captured at Challenge #6 A > xi and click on "Check out" button

h. Once checked out, validate that the book name, barcode number is displayed just below the "Check out" button. Note down the value of "Due date" from the success message.

i. Once checked out, validate that the book details are visible under "Checkouts" tab and the number next to checkouts should be equal to 1 or greater than 1.

8. **B) Switch to OPAC UI Interface and validate that the book is now issued to patron**

   a. Switch to OPAC UI & navigate to "Welcome, User > Account > Summary"

   b. Validate that the book is now checked out for this patron.

   c. Validate Barcode number & Due Date that was captured on Challenge #8 A > h

   d. Click on "Checkout History" and validate that the book that was issues is visible here along with due date

   e. Search for the same book and validate that the holdings tab for this book should display the record as "Checked out" along with "Date due" equals to the once captured on Challenge #8 A > h

9 A) and 9 B) is multi browser have to be done together – total marks : 20

9. **A) Using Koha API, add credits to the currently logged in patron**

   a. Add Credit to patron's account using patron API (Koha REST API)

   b. The value of credit should be random number between 100-500

   c. Validate the response should have the same value that was used to set credit amount for the patron

9. **B) Switch to OPAC UI / Staff Interface & validate that the credit amount is reflected for the patron**

   a. Switch to OPAC UI > Open "Charges" section under patron account and validate that the same amount is displayed

   b. Switch to Staff UI interface and on Patron details page, validate that the same credit value is displayed in "Attention" section

10. **Submit or check in the book in library from Staff UI interface - 10 MARKS**

    a. Switch to Staff interface > Circulation > Check in

    b. Enter the barcode number that was used to "Check out" the book/item in Challenge #8 A > g and click on "Check in" button

    c. On the "Checked-in items" table, validate that the book that was issues is now checked in to library. Validate book details from the table

11. **Switch to OPAC UI & validate that the book is no longer issued to patron – 10 Marks**
    a. Switch to OPAC UI and open Summary page
    b. Validate that the book is no longer visible under “Checked out” section
    c. Open “Checkout History” page and validate the status of the book is now “Checked out”

12. **On OPAC UI, search for the book and validate holdings availability – 10 marks**
    a. Search for the same book from OPAC Ui and navigate to book details page
    b. On Holding tab, validate that the book status should now state “Available”

13. **Book cover image comparison in OPAC UI – 30 Marks**
    a. Login on OPAC UI with valid patron
    b. Search for a book “THE NAME OF BOOK TO SEARCH”
    c. Capture the screenshot of book cover
    d. Search for another book “THE NAME OF BOOK WITH DIFFERENT EDITION”
    e. Capture the screenshot of the book cover
    f. Store both the screenshot in report as well as in code repository
    g. The pattern of the file should be “name-of-the-book-compare-yyymmdd-hhmmss.png”
    h. Compare the images and assert that both are same or not

Challenge B

Hybrid Scenario: Desktop & Web Automation for Koha

Download & Install MarcEdit desktop application:

Windows	https://www.mediafire.com/file/zru3yeb78kaf269/MarcEdit_7_7_User_Install.exe/file
MacOs	https://www.mediafire.com/file/bdkttmxxt965w8e/MarcEdit3_5.pkg.zip/file

Once installed, download MARC file based on your team number from following link.

(Please ask JURY member for your MARC files)

1: Desktop app automation – 50 marks

1. Open MarcEdit desktop application using your choice of automation tool
2. Go to "Tools > MARC Processing Tools > MARCSplit"
3. On the new window,
    a. at field "Source", select the MARC file that you downloaded earlier.
    b. At field "Destination", select the folder location where you want to export the files
    c. For Options,
        - Records per file: 10
        - Check the checkbox against label "# of files" and enter "10"
    d. Click "Process"
    e. Validate the message "10 files have been generated"
    f. Click on "Close" button
4. Close MarcEdit application

2: (Web app automation) – 30 Marks

1. Open Koha Staff UI, login to it using Admin credentials
2. Navigate to "Cataloging > Stage records for import > Choose File"
    a. Select any files from the files that was exported in Challenge #1 > 3 > b
    b. Click on "Upload file"
    c. Wait until upload progress reads "100%". *This needs to be validated and dynamic wait must be used*
    d. Leave all the details of the form as is and click on "Stage for import" button visible at the bottom of the page
    e. Wait until the job has been completed, on this page, validate following
        i. the status on progress bar should read "100% finished"

- ii. Dynamic wait must be used for waiting mechanisms. No hard wait is acceptable

f. Once completed, click on “View batch” link

g. On the “**Manage staged MARC records”** screen, validate following
   - i. Type should be “Bibliographic records”
   - ii. Name of the file should be the one that you uploaded
   - iii. Staged filed should have Today’s date
   - iv. Status should be “Staged”

h. For following two fields on the form, select option as “Books, Booklets, Workbooks”
   - i. New record framework:
   - ii. Replacement record framework:

i. In the table displayed at the bottom of the page, you should see list of books that will be imported as a part of this process

j. Copy the values from following columns and store it in a file of your choice
   - i. #
   - ii. Citation
   - iii. Status
   - iv. Match Type

k. Click on “Import this batch into the catalog”

l. Once clicked, you should see job page.
   - i. Validate message “**The job has been enqueued!”**

m. Once refreshed, implement the logic for following
   - i. Refresh the page until “Status” field reads “Finished”
   - ii. Print the values of the following fields
     1. Job ID
     2. Status
     3. Progress
     4. Type
     5. Queued
     6. Started
     7. Ended
   - iii. Validate that the date value of “Ended” field is greater than date value of “Started” field
   - iv. In “Report” section, validate message “Completed import of records”
   - v. Save the details of the report table in a file.

3: Koha API – 20 Marks

Rules:

- To complete this challenge, you will need to read the file that was saved in “Challenge #2 > 2 > j”

- This test should be data driven. Meaning it should read the file and process records one after another and implement following challenge

1. Iterate through each record and search for the imported book using /biblios API endpoint and validate following
    a. Status code
    b. The book details
    c. Response schema validation
    d. Assert the title should match expected value

    Note: To understand how filtering works please review the docs here (https://api.koha-community.org/24.05.html#section/Filtering-responses)

2. Store the json response in as a json file in code repository
    - The pattern of the json file should be "biblio_id-title-with-space-as-comma-separated-and-in-lower-case.json"