Trello Workspaces Recent More SP

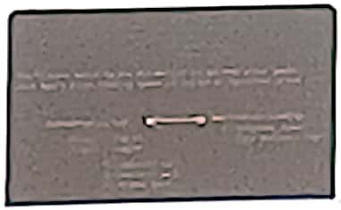
What is Synchronization?

in list Synchronization

Description

Add a more detailed description...

Attachments



Synchronization

Added Feb 6 at 12:39 AM - Comment - Delete - Edit

Make cover

Suggested

Join

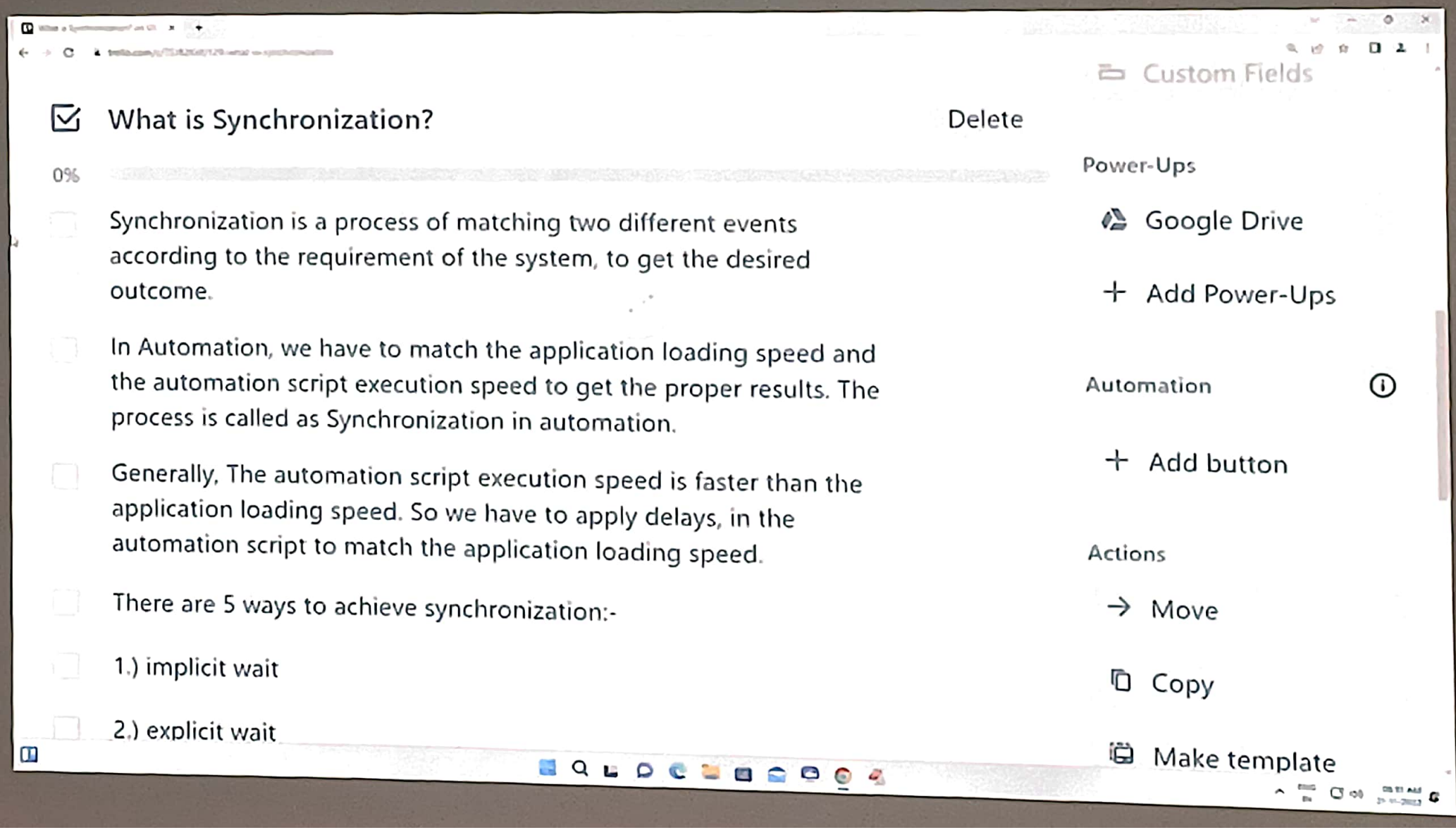
Add to card

Members

Labels

Checklist

Dates


What is Synchronization?
Delete
0%
Synchronization is a process of matching two different events according to the requirement of the system, to get the desired outcome.
In Automation, we have to match the application loading speed and the automation script execution speed to get the proper results. The process is called as Synchronization in automation.
Generally, The automation script execution speed is faster than the application loading speed. So we have to apply delays, in the automation script to match the application loading speed.
There are 5 ways to achieve synchronization:-
1.) implicit wait
2.) explicit wait
Custom Fields
Power-Ups
Google Drive
Add Power-Ups
Automation
Add button
Actions
Move
Copy
Make template

1.) implicit wait

2.) explicit wait

3.) Fluent wait.

4.) Thread.sleep()

5.) pageloadTimeout()

Note:

Application Loading Speed will be slow due to:

1.) Slow Internet Connection.

2.) Poor Business logics written in the webserver.

Add an item

Copy

Make template

Watch

Archive

Share

What is implicit wait?

in list Synchronization

Description Edit

Earlier Signature:

```
public TimeOuts implicitlyWait(Long timeDelay, TimeUnit.CONSTANT) {
}
```

Latest from and after 4 version Signature:

```
public Duration implicitlyWait(Duration duration) {
}
```

Suggested

Join

Add to card

Members

Labels

Checklist

Dates

Attachment

Latest from and after 4 version Signature:

public Duration implicitlyWait(Duration duration) {

}

Earlier Usage:

driver.manage().timeouts().implicitlyWait(15, TimeUnit.SECONDS);

Current Usage:

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

Attachments

Labels

Checklist

Dates

Attachment

Cover

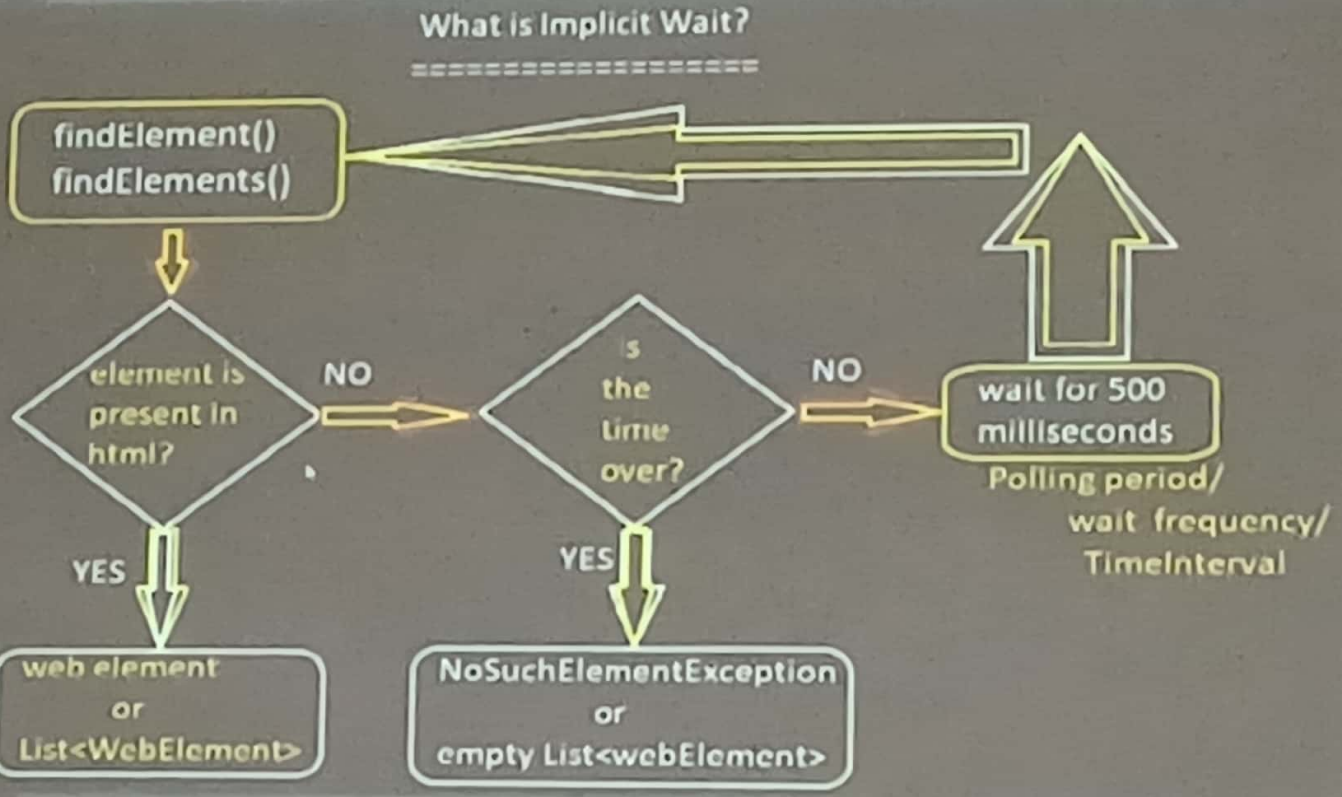
Custom Fields

Power-Ups

Google Drive

Add Power-Ups

Automation



What is Implicit Wait?

Added Feb 2 at 3:22 PM - 29.92 KB

Open in new tab Make cover Delete

About:

Delete

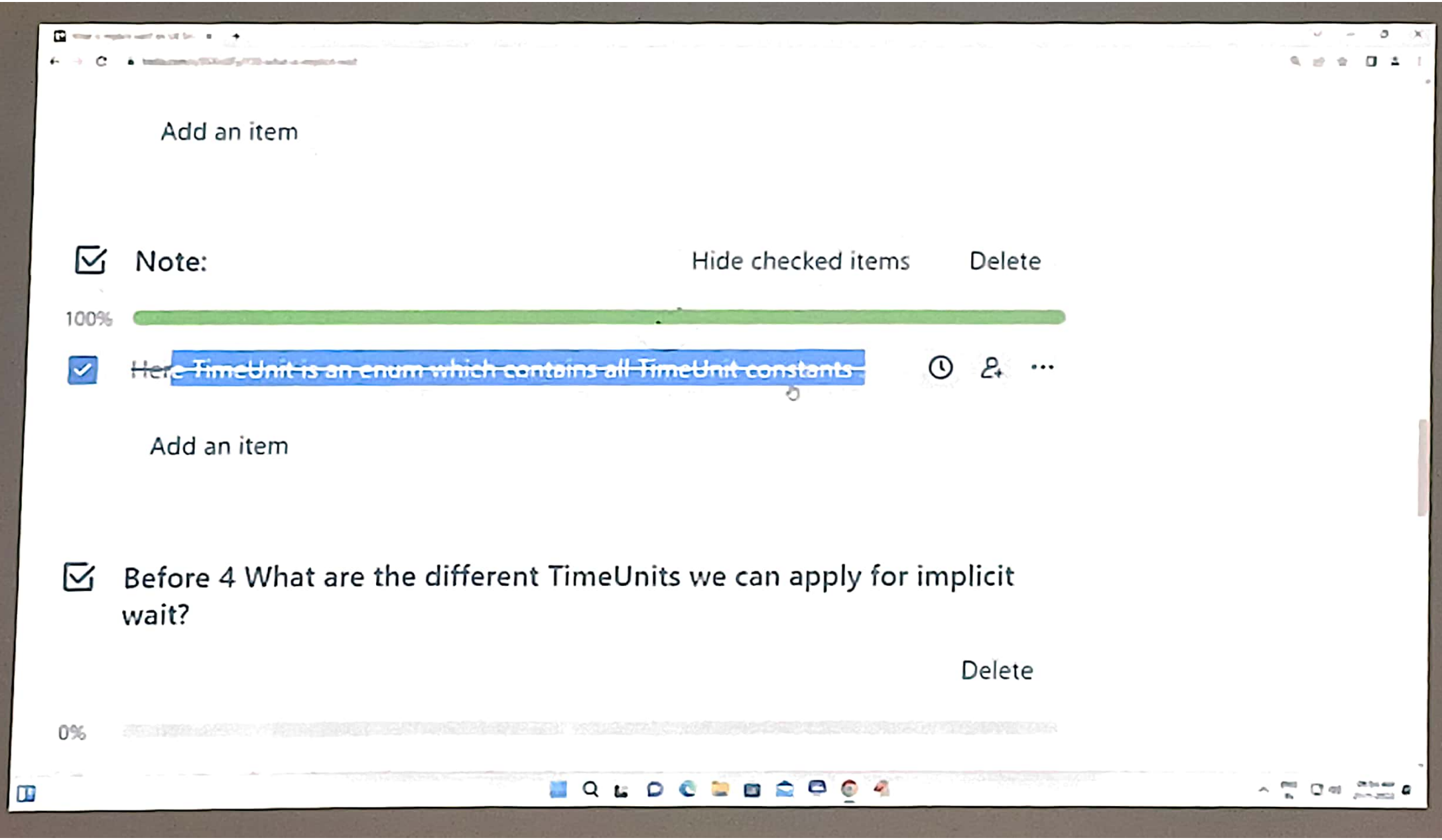
Archive

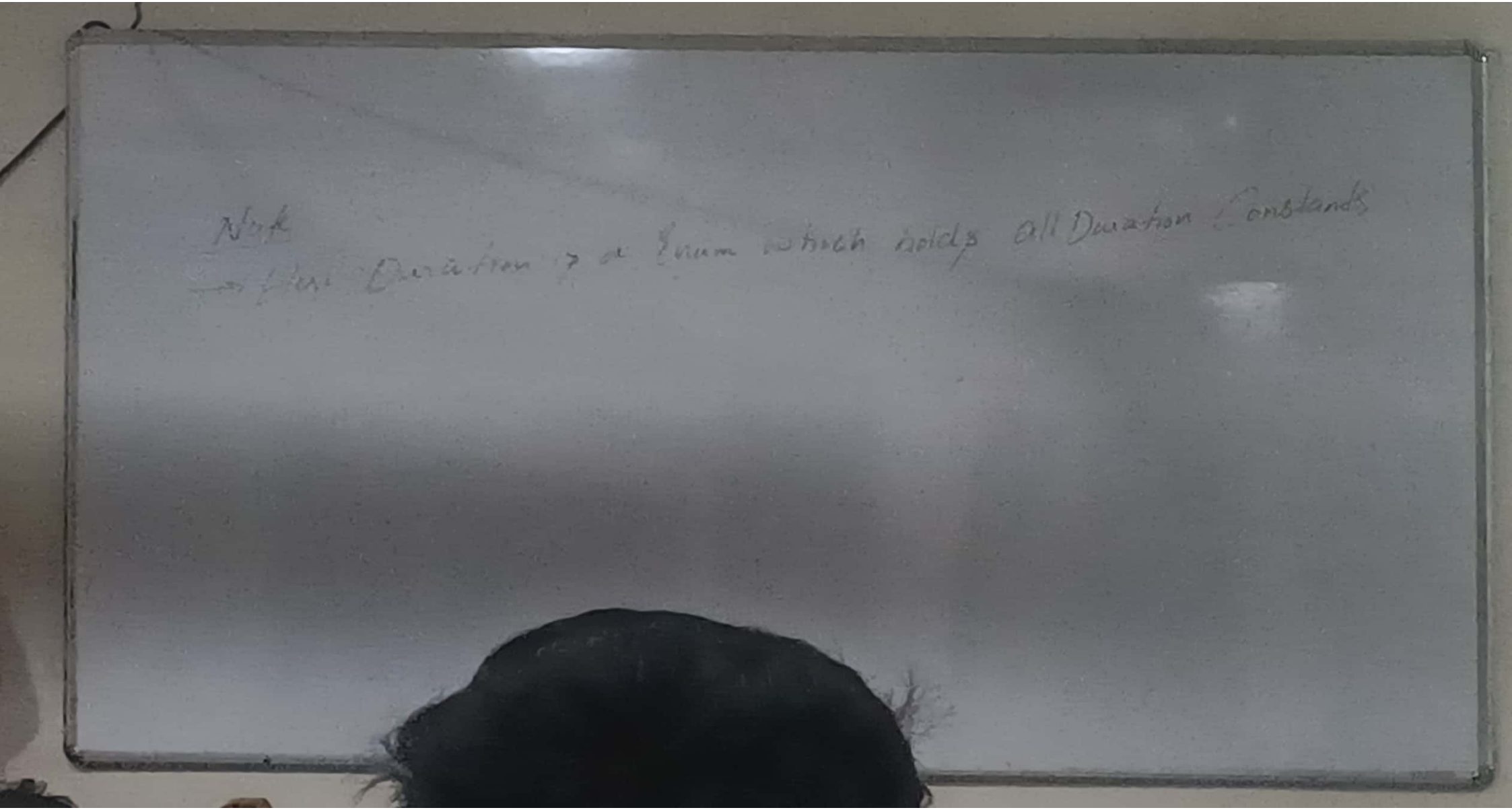
Share

0%

- [ ] It's one of the Intelligent waits in selenium WebDriver which helps us to save time while we wait for the WebElements to load on the webpage/HTML/DOM(Document Object Model).
- [ ] Here we provide the time delay for searching elements in the webpage.
- [ ] If the element is found anytime within the specified time delay then the code doesn't wait for the remaining duration. It returns the found webelement. So we call it has intelligent wait.
- [ ] It uses 500 milliseconds of polling period or the search frequency.
- [ ] If the element is not found within the specified delay then we will get NoSuchElementException (by findElement()) or empty List<WebElement> (by findElements()).
- [ ] ****This wait is applicable only for 2 methods findElement() and findElements().

Add an item


Add an item
Note:
Hide checked items
Delete
100%
Here TimeUnit is an enum which contains all TimeUnit constants
Add an item
Before 4 What are the different TimeUnits we can apply for implicit wait?
Delete
0%


Note
→ Java Duration is a Enum which holds all Duration Constants

Before 4 What are the different TimeUnits we can apply for implicit wait?

Delete

0%

- [ ] We can apply 7 TimeUnit beyond version 4:
- [ ] TimeUnit.SECONDS - ofSeconds(long seconds);
- [ ] TimeUnit.MILLISECONDS - ofMillis(long millis);
- [ ] TimeUnit.MICROSECONDS -- xx
- [ ] TimeUnit.NANOSECONDS - ofNanos(long nanos);
- [ ] TimeUnit.MINUTES - ofMinutes(long minutes);
- [ ] TimeUnit.HOURS - ofHours(long hours);

We can apply 7 TimeUnit beyond version 4:

TimeUnit.SECONDS - ofSeconds(long seconds);

TimeUnit.MILLISECONDS - ofMillis(long millis);

TimeUnit.MICROSECONDS -- xx

TimeUnit.NANOSECONDS - ofNanos(long nanos);

TimeUnit.MINUTES - ofMinutes(long minutes);

TimeUnit.HOURS - ofHours(long hours);

TimeUnit.DAYS - ofDays(long days);

Note: The usage of the method implicitlyWait(Long timeDelay, TimeUnit.CONSTANT) is depreciated in selenium 4 and above version instead of that we should use implicitlyWait(Duration duration).

Note

→ Here Duration is a Enum which holds all Duration Constants

→ Duration of Seconds( )

→ Duration of Minutes( )

→ Duration of Nanoseconds

→ Duration of Milliseconds

→ Duration of Hours( )

→ Duration of Days( )

TimeUnit.CONSTANT) is depreciated in selenium 4 and above version instead of that we should use implicitlyWait(Duration duration).

Add an item

**NOTE:

Delete

0%

- [ ] This implict timeout is applicable for each and every call of findElement() and findElements(). So generally it will be applied at the beginning of the automation script as a Pre-condition.

Add an item

Activity

Show details

What is explicit wait?

in list Synchronization

Description

Add a more detailed description...

Attachments

What is explicit wait? ↗

Added Feb 2 at 5:02 PM - Comment - Delete - Edit

Make cover

Suggested

Join

Add to card

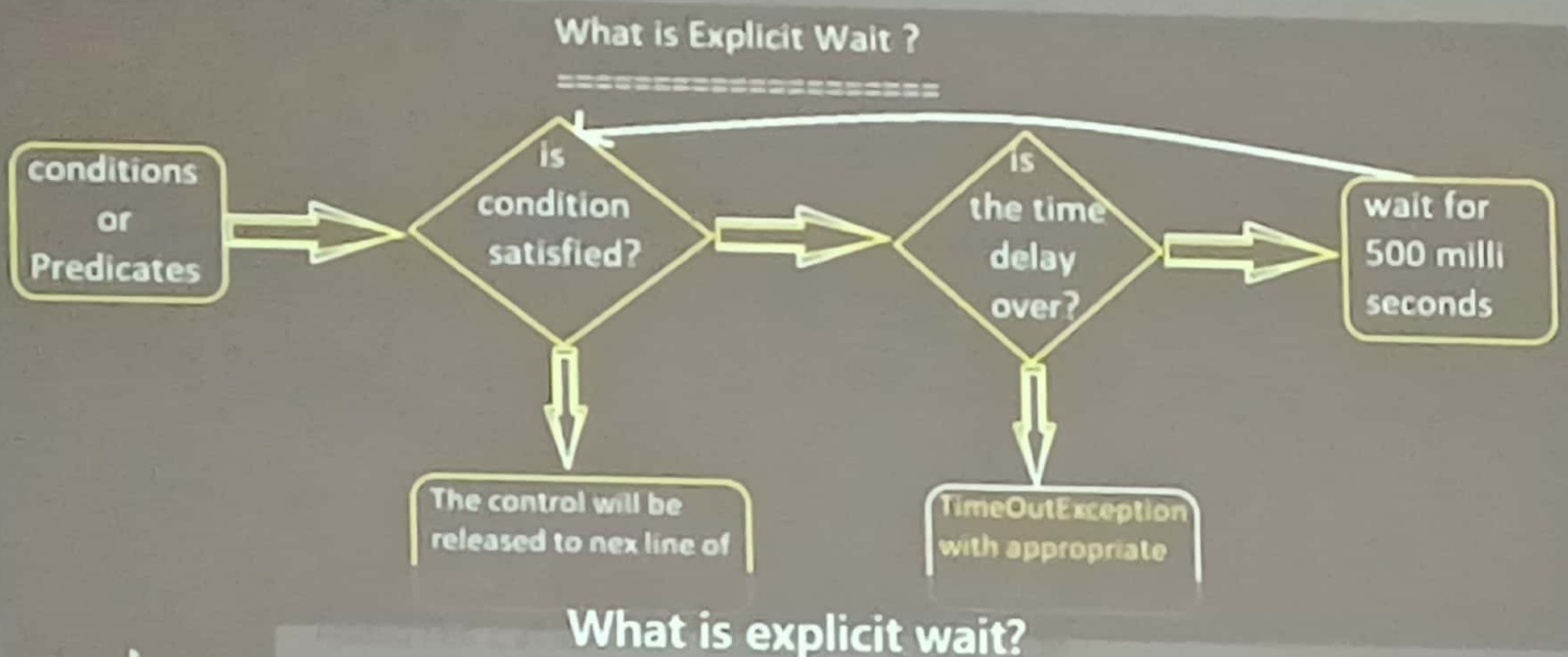
Members

Labels

Checklist

Dates

Attachment



What is explicit wait?

Added Feb 2 at 5:02 PM - 27.16 KB

Open in new tab Make cover Delete

About:

Delete

Make template

Watch

Archive

Share

0%

It is one of the intelligent waits in Selenium Webdriver which helps us to save time while we wait for:

1.)Expected title of the webpage to load.

2.)Expected URL of the webpage to load.

3.)Element to reach the clickable state.

4.)Image to be visible on webpage.

5.)Alert to be present on the webpage and many more..!

Here we provide the time delay for the expected conditions to be sucessfull.

If the expected condition is satisfied anytime within the specified time delay then the code doesn't wait for the remaining duration. It

- [ ] Here we provide the time delay for the expected conditions to be sucessfull.
- [ ] If the expected condition is satisfied anytime within the specified time delay then the code doesn't wait for the remaining duration. It releases the control to the next line of automation script.
- [ ] It uses 500 milliseconds of polling period or the search frequency.
- [ ] If the given condition is not successful within the specified delay then we will get TimeOutException with the appropriate error message.

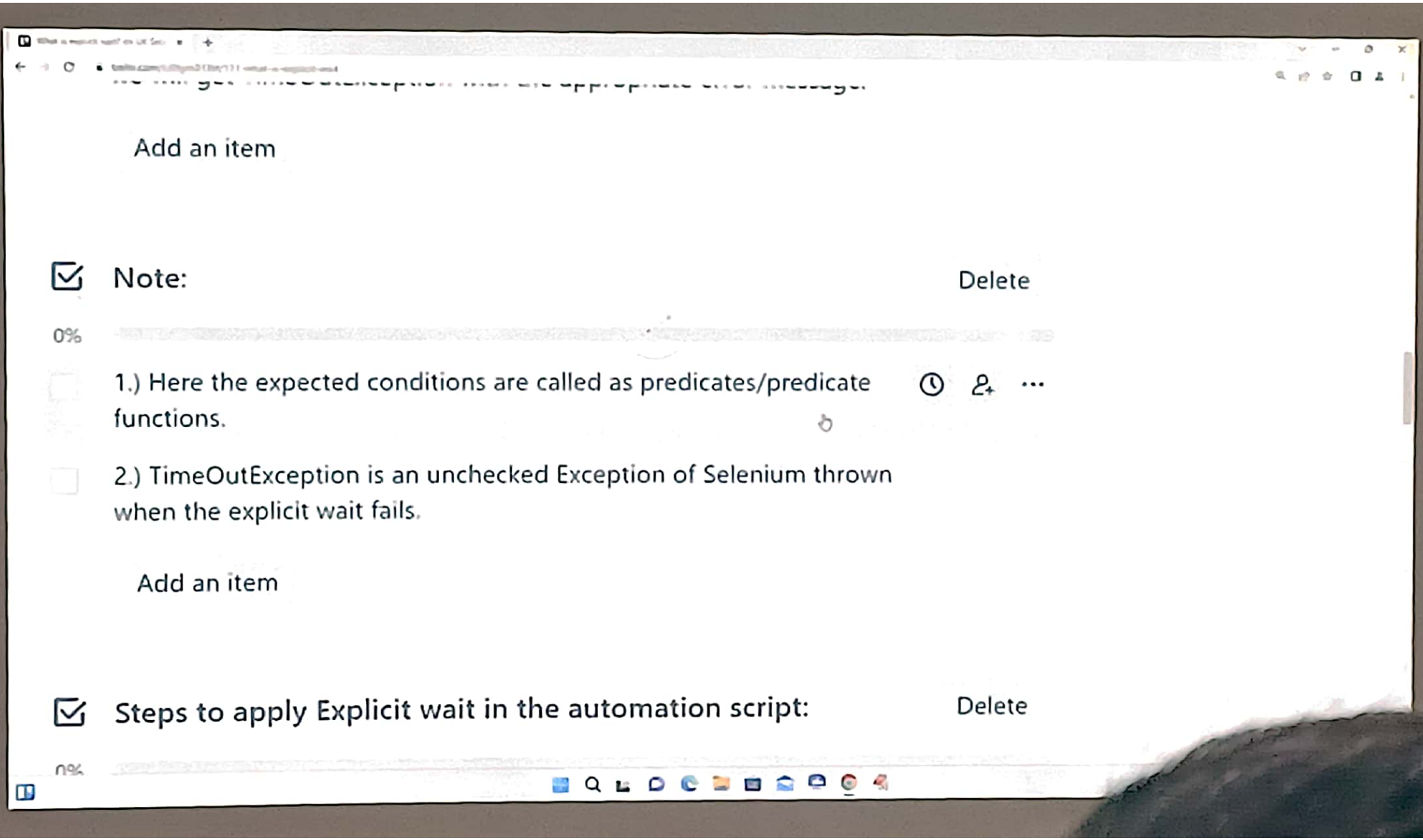
Add an item

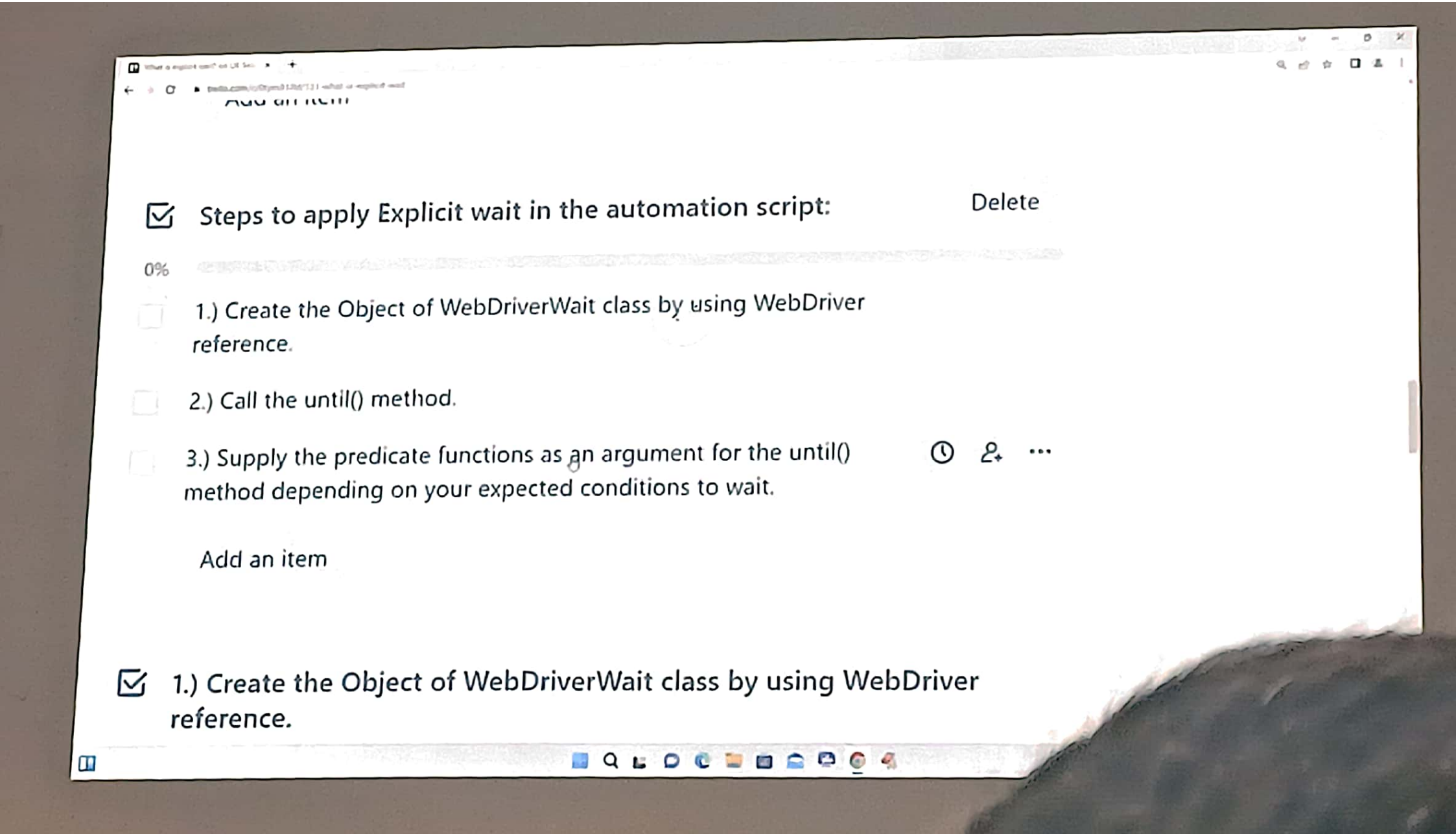
Note:

Delete

0%

- [ ] 1.) Here the expected conditions are called as predicates/predicate


Add an item
Note:
Delete
0%
1.) Here the expected conditions are called as predicates/predicate functions.
2.) TimeOutException is an unchecked Exception of Selenium thrown when the explicit wait fails.
Add an item
Steps to apply Explicit wait in the automation script:
Delete


Add an item
Steps to apply Explicit wait in the automation script:
Delete
0%
1.) Create the Object of WebDriverWait class by using WebDriver reference.
2.) Call the until() method.
3.) Supply the predicate functions as an argument for the until() method depending on your expected conditions to wait.
Add an item
1.) Create the Object of WebDriverWait class by using WebDriver reference.

☑ 1.) Create the Object of WebDriverWait class by using WebDriver reference.

Delete

0%

- [ ] There is built-in class called WebDriverWait in Selenium package.
- [ ] It is a Concrete class.
- [ ] The Constructor of WebDriverWait receives two arguments:
- [ ] 1.) WebDriver Reference
- [ ] This represents the browser and application on which explicit wait has to be applied.
- [ ] 2.) The time delay in Seconds.
- [ ] The time delay for the explicit wait conditions.

- [ ] The Constructor of WebDriverWait receives two arguments:

- [ ] 1.) WebDriver Reference

- [ ] This represents the browser and application on which explicit wait has to be applied.

- [ ] 2.) The time delay in Seconds.

- [ ] The time delay for the explicit wait conditions.

- [ ] Ex: WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(20));

Add an item

- [x] 2.) Call the until() method

Hide checked items Delete

25%

2.) Call the until() method

Hide checked items Delete

25%

- [ ] until() is a non-static method of WebDriverWait class, whenever we have multiple conditions then we can call this method again and again based on the conditions.
- [ ] For each and every until() invocation the overall time delay is applied.
- [ ] WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(20));
- [x] ~~Refer the above diagram.~~

Add an item

3.) Supply the predicate functions as an argument for the until() method depending on your expected conditions to wait.

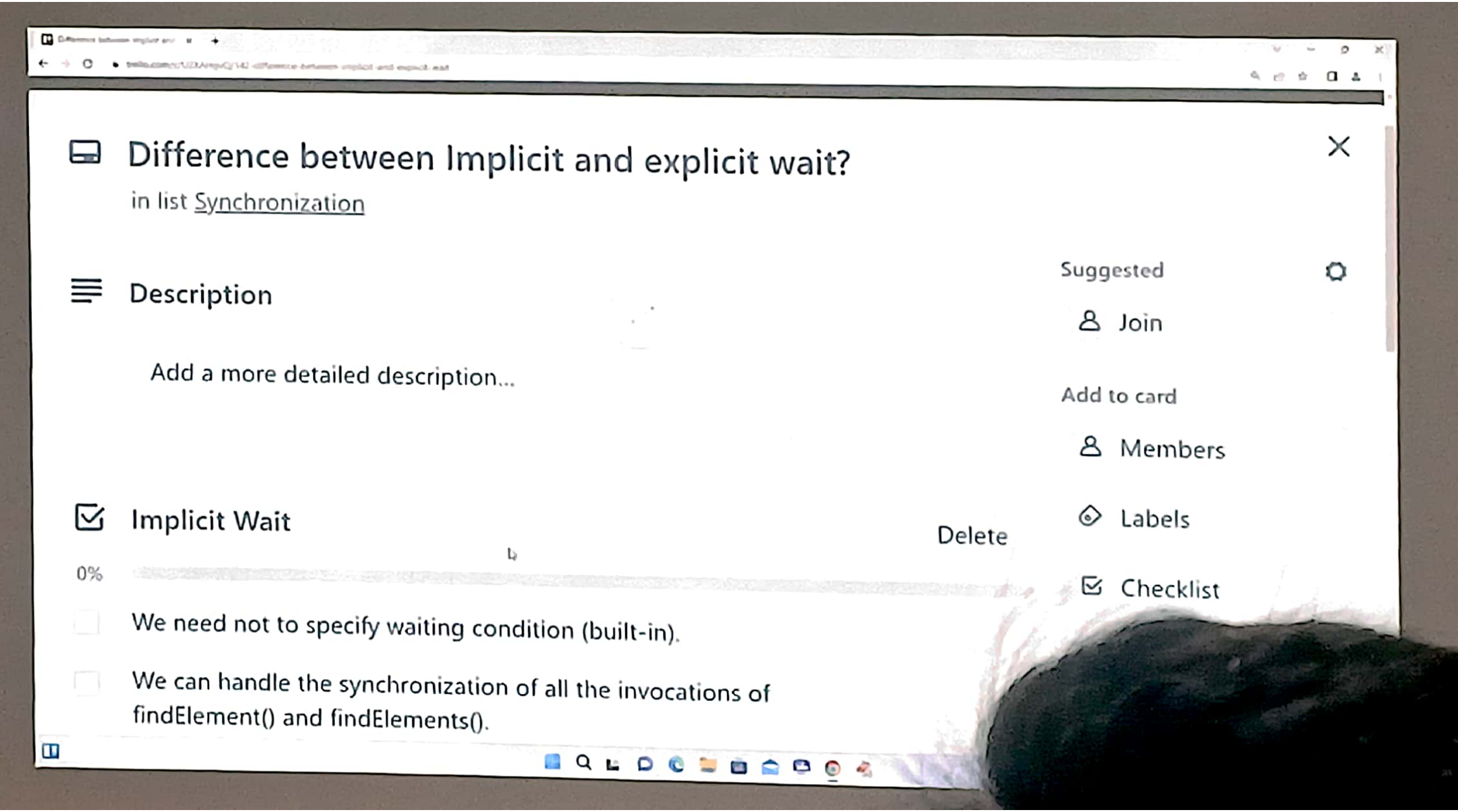
Add an item

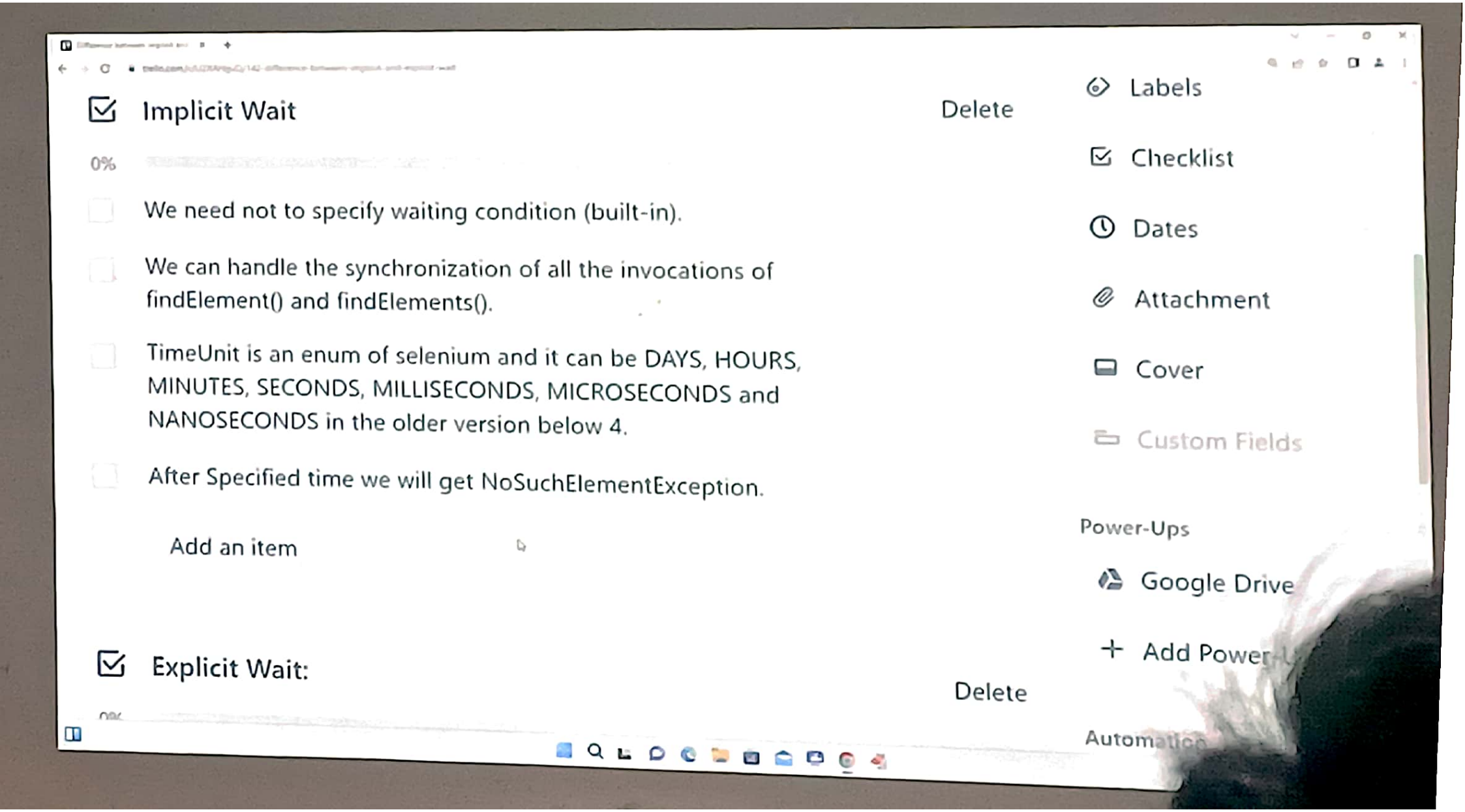
- [x] 3.) Supply the predicate functions as an argument for the until() method depending on your expected conditions to wait.

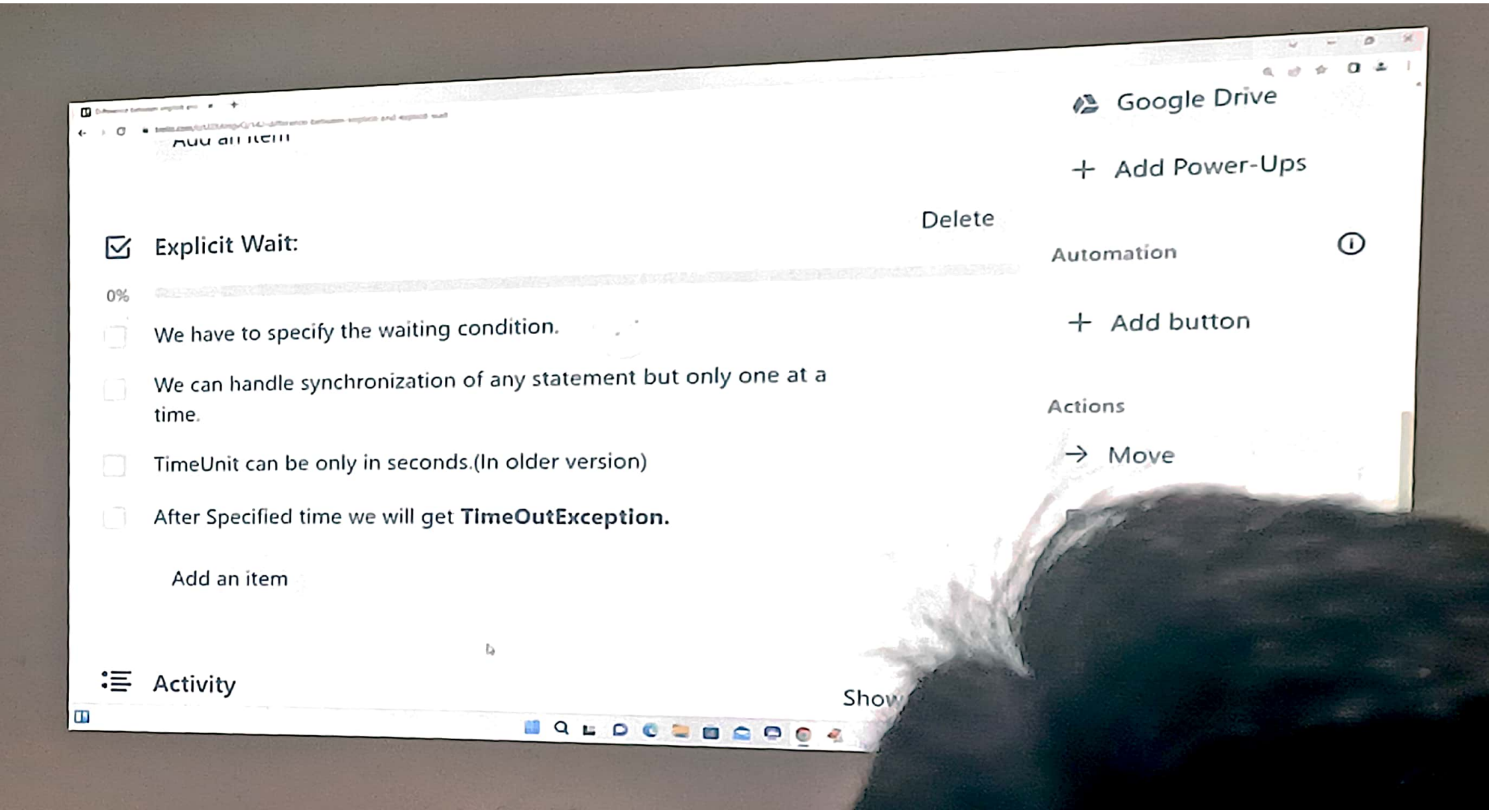
. Hide checked items Delete

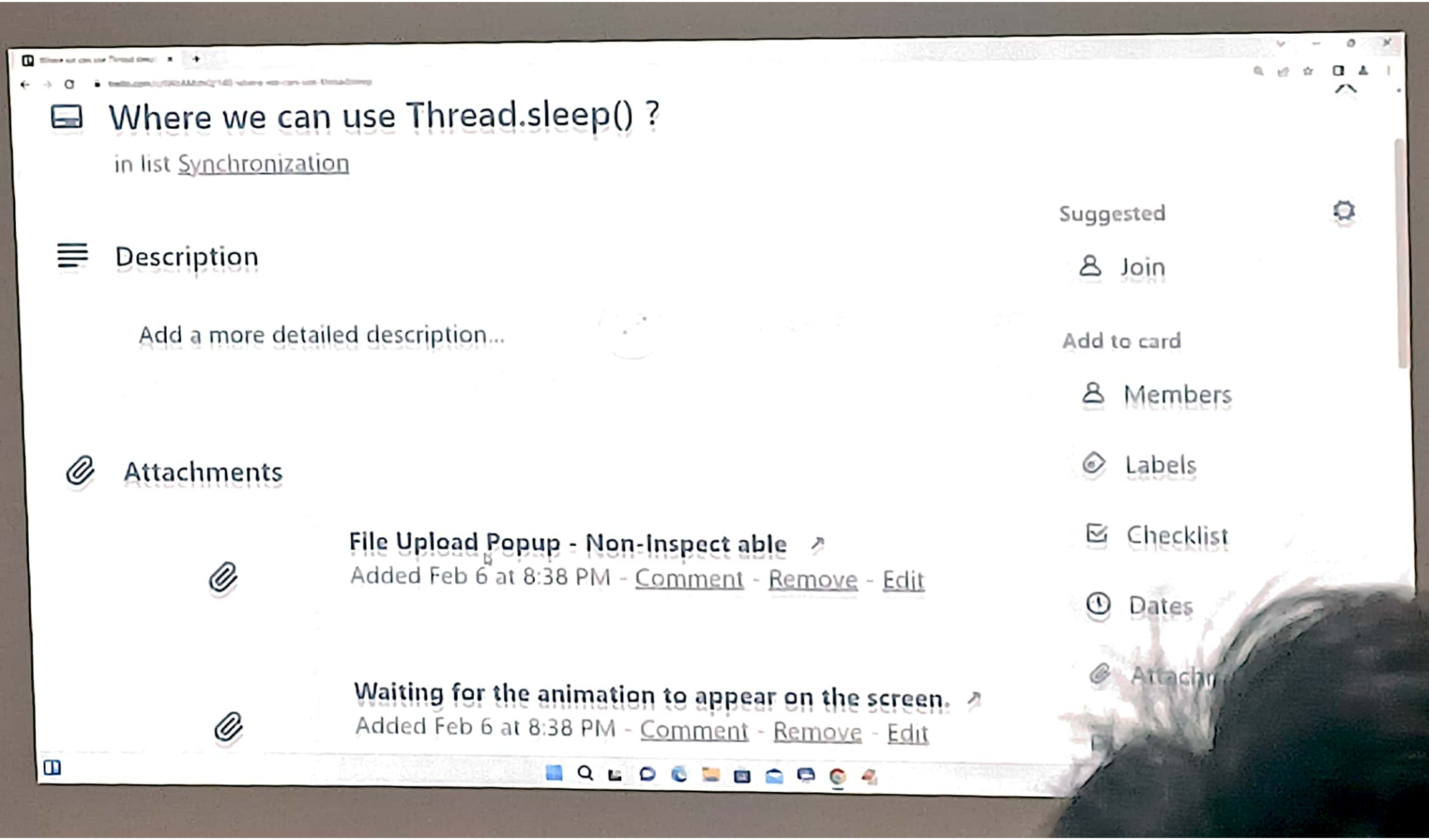
20%

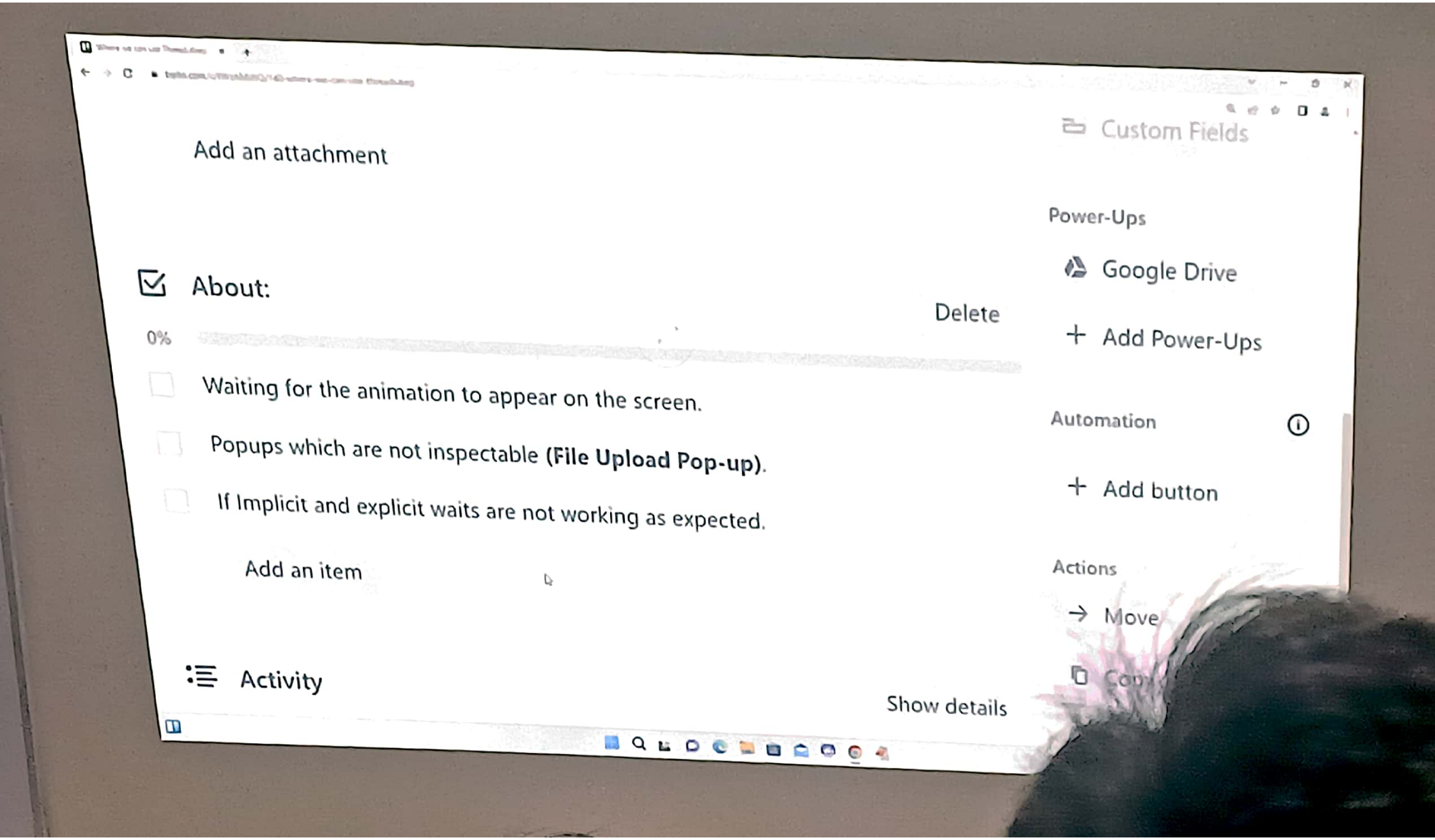
- [x] ~~Refer the above diagram.~~
- [ ] ExpectedConditions: It is the concrete class of Selenium and it is having of lot of static methods, which were also known as predicate functions.
- [ ] Depending on the conditions, we have to use the various static methods of this class.
- [ ] **NOTE:** If the static Methods were not their as the condition to specify, then we can create our own predicate functions by using Lambda Expressions

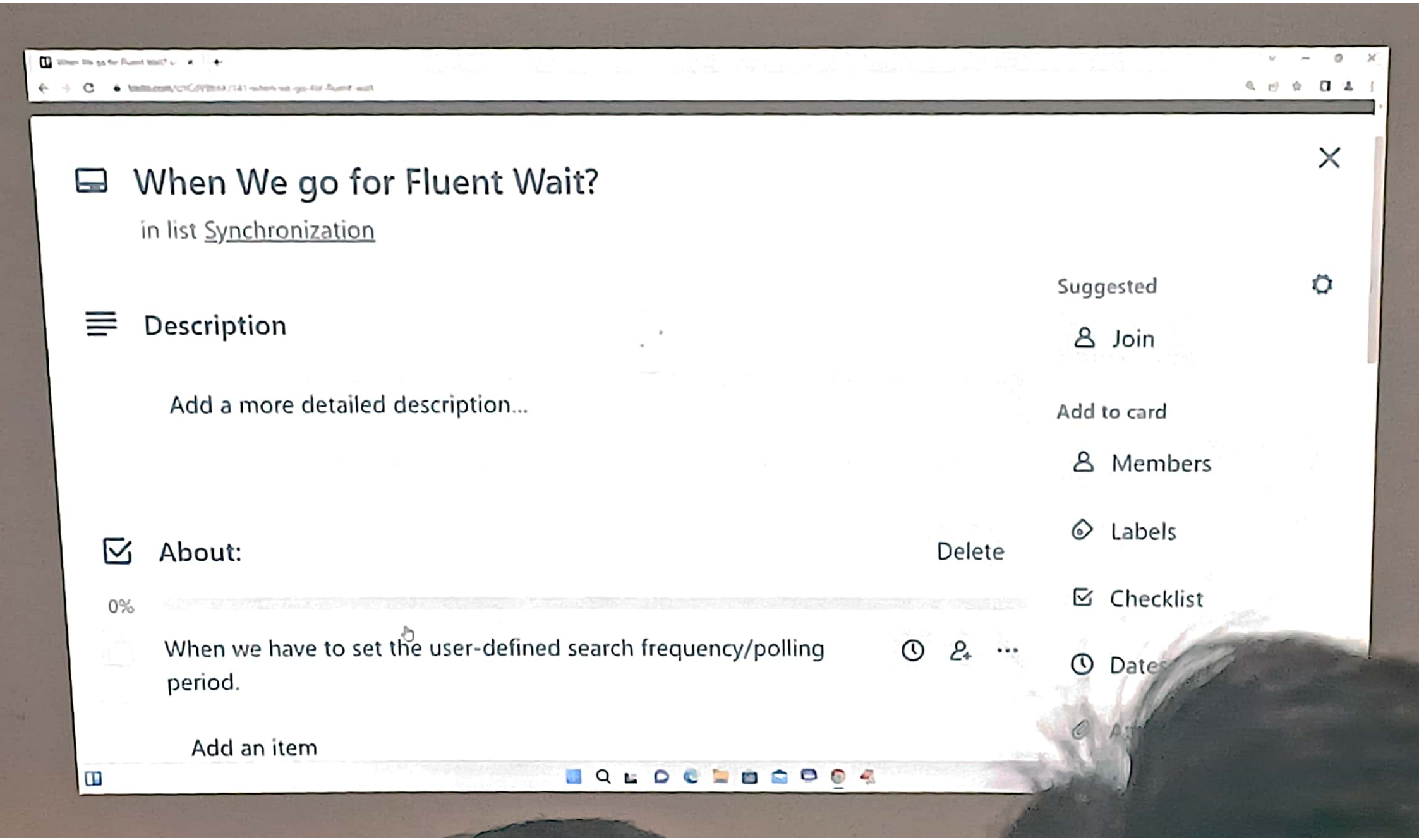

Difference between Implicit and explicit wait?
in list Synchronization
Description
Add a more detailed description...
Implicit Wait
Delete
0%
We need not to specify waiting condition (built-in).
We can handle the synchronization of all the invocations of findElement() and findElements().
Suggested
Join
Add to card
Members
Labels
Checklist

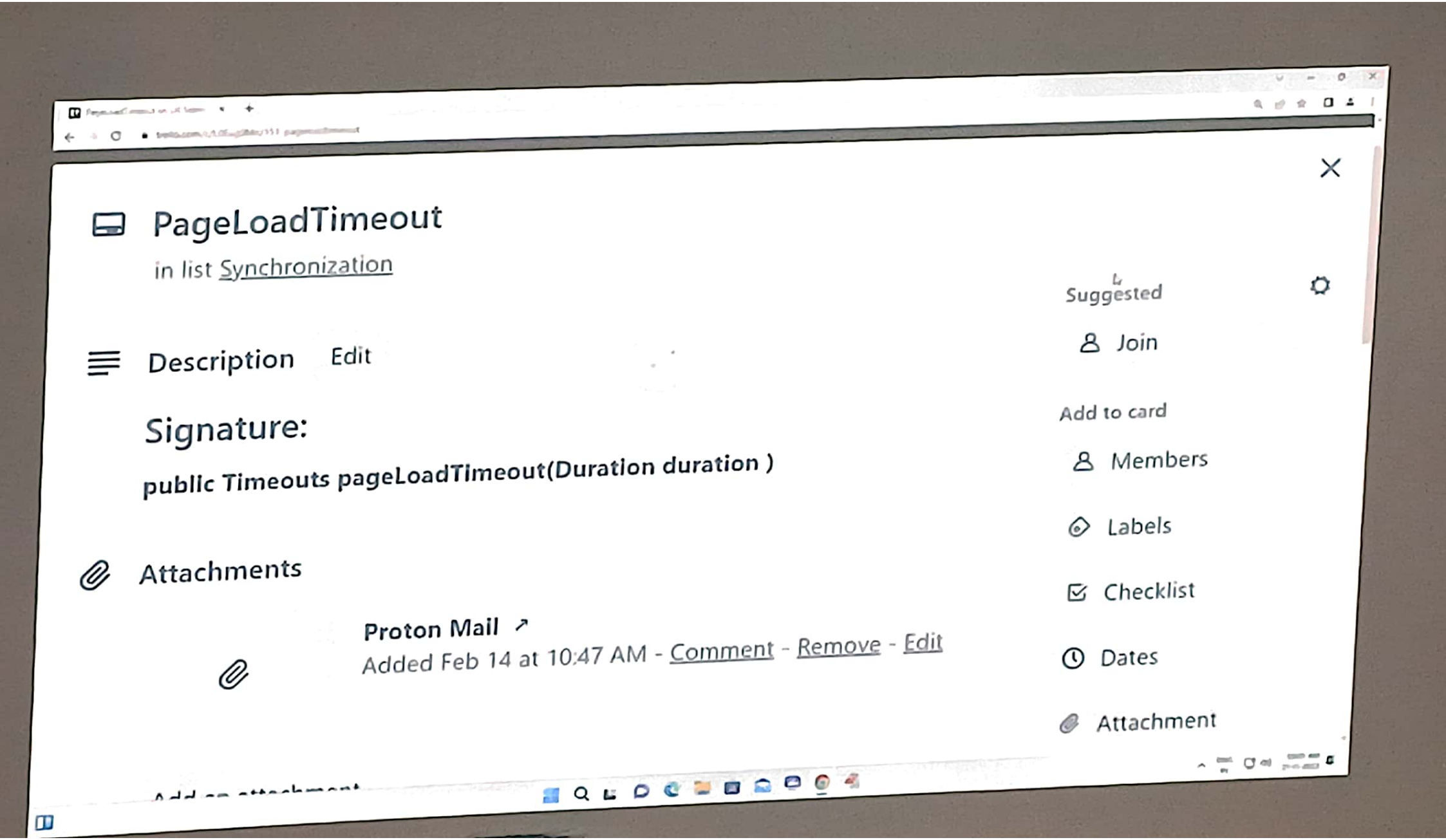

Implicit Wait
Delete
0%
We need not to specify waiting condition (built-in).
We can handle the synchronization of all the invocations of findElement() and findElements().
TimeUnit is an enum of selenium and it can be DAYS, HOURS, MINUTES, SECONDS, MILLISECONDS, MICROSECONDS and NANOSECONDS in the older version below 4.
After Specified time we will get NoSuchElementException.
Add an item
Explicit Wait:
Delete
Labels
Checklist
Dates
Attachment
Cover
Custom Fields
Power-Ups
Google Drive
Add Power-Up
Automation


Google Drive
Add Power-Ups
Delete
Explicit Wait:
Automation
0%
Add button
We have to specify the waiting condition.
We can handle synchronization of any statement but only one at a time.
Actions
TimeUnit can be only in seconds.(In older version)
Move
After Specified time we will get TimeOutException.
Add an item
Activity


Where we can use Thread.sleep() ?
in list Synchronization
Description
Add a more detailed description...
Attachments
File Upload Popup - Non-Inspect able
Added Feb 6 at 8:38 PM - Comment - Remove - Edit
Waiting for the animation to appear on the screen.
Added Feb 6 at 8:38 PM - Comment - Remove - Edit
Suggested
Join
Add to card
Members
Labels
Checklist
Dates


Add an attachment
Custom Fields
Power-Ups
Google Drive
About:
Delete
Add Power-Ups
0%
Waiting for the animation to appear on the screen.
Automation
Popups which are not inspectable (File Upload Pop-up).
Add button
If Implicit and explicit waits are not working as expected.
Add an item
Actions
Move
Activity
Show details


When We go for Fluent Wait?
in list Synchronization
Description
Add a more detailed description...
About:
Delete
0%
When we have to set the user-defined search frequency/polling period.
Add an item
Suggested
Join
Add to card
Members
Labels
Checklist


PageLoadTimeout
in list Synchronization
Suggested
Join
Description
Edit
Signature:
public Timeouts pageLoadTimeout(Duration duration)
Add to card
Members
Labels
Checklist
Dates
Attachment
Attachments
Proton Mail
Added Feb 14 at 10:47 AM - Comment - Remove - Edit

0%

- [ ] It is Applicable only for **get()** and **navigate().to()**;
- [ ] By Default both the methods has set for infinite time delay.

Add an item

About

Delete

0%

- [ ] **PageLoadTimeOut** it is the amount Of time set to load the webpage in the browser which has opened in the current session by the script.
- [ ] It will help us to verify whether the webpage can be loaded within the given time or not.
- [ ] Further loading of the application will be freezed after the given time in the **pageLoadTimeout** has got over.

Power-Ups

Google Drive

+ Add Power-Ups

Automation

+ Add button

Actions

Move

Copy

Make template

Watch

0%

- [ ] It is Applicable only for **get()** and **navigate().to()**;
- [ ] By Default both the methods has set for infinite time delay.

Add an item

About

Delete

0%

- [ ] **PageLoadTimeOut** it is the amount Of time set to load the webpage in the browser which has opened in the current session by the script.
- [ ] It will help us to verify whether the webpage can be loaded within the given time or not.
- [ ] Further loading of the application will be freezed after the given time in the **pageLoadTimeout** has got over.

Power-Ups

Google Drive

+ Add Power-Ups

Automation

+ Add button

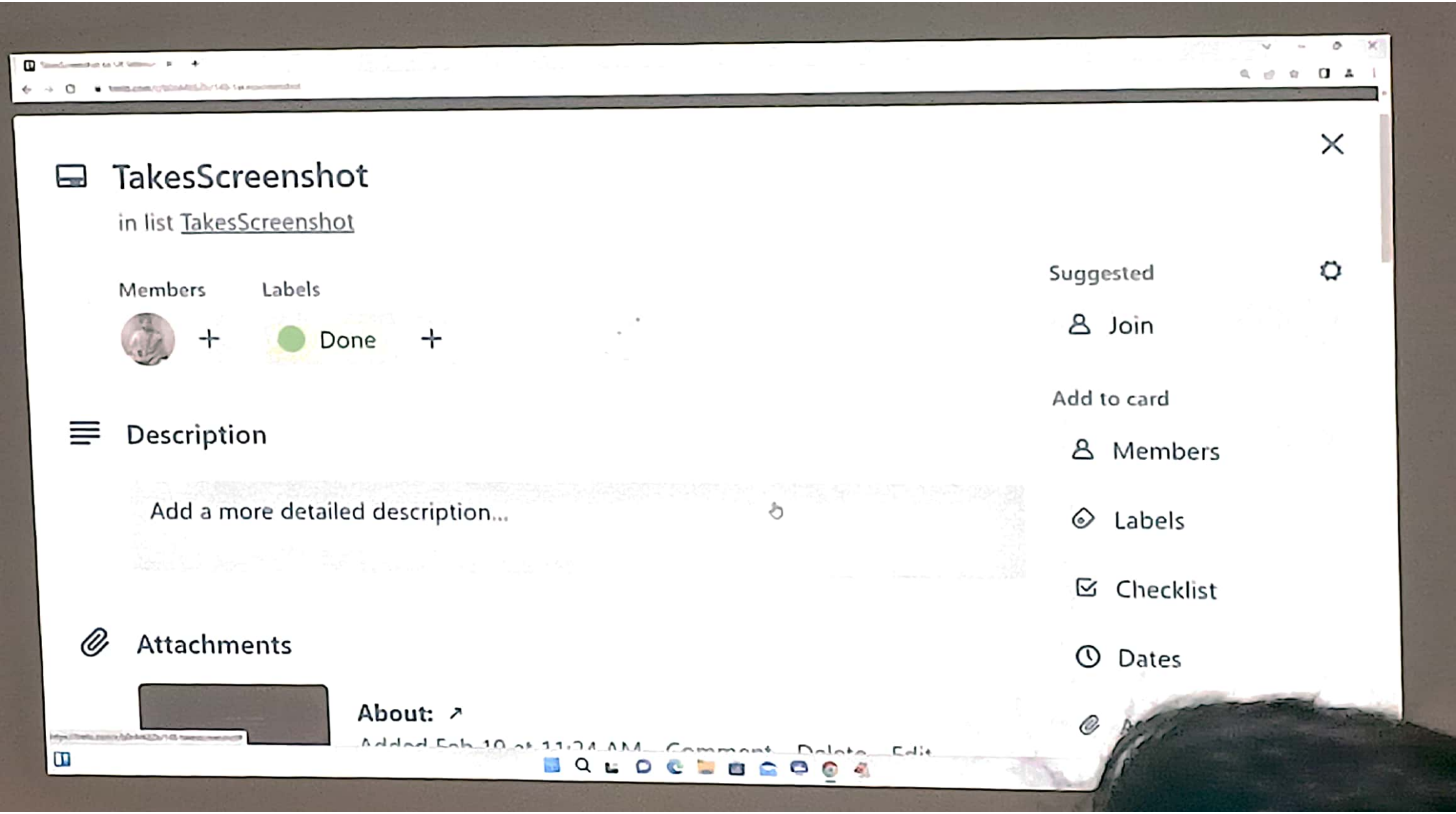
Actions

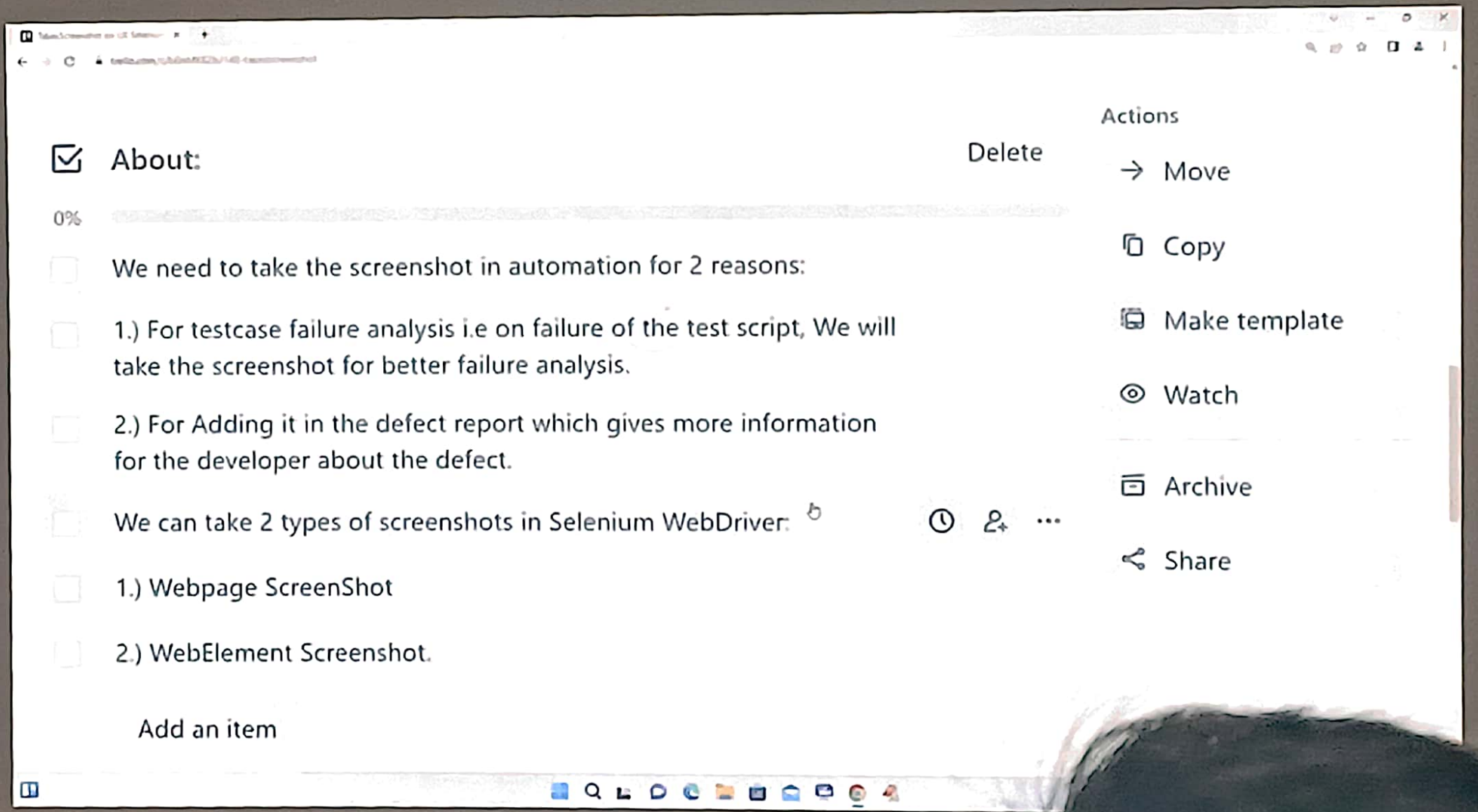
→ Move

Copy

Make template

Watch


TakesScreenshot
in list TakesScreenshot
Members
Labels
Done
Suggested
Join
Add to card
Members
Labels
Checklist
Dates
Description
Add a more detailed description...
Attachments
About:


About:
Delete
0%
We need to take the screenshot in automation for 2 reasons:
1.) For testcase failure analysis i.e on failure of the test script, We will take the screenshot for better failure analysis.
2.) For Adding it in the defect report which gives more information for the developer about the defect.
We can take 2 types of screenshots in Selenium WebDriver:
1.) Webpage ScreenShot
2.) WebElement Screenshot.
Add an item
Actions
Move
Copy
Make template
Watch
Archive
Share

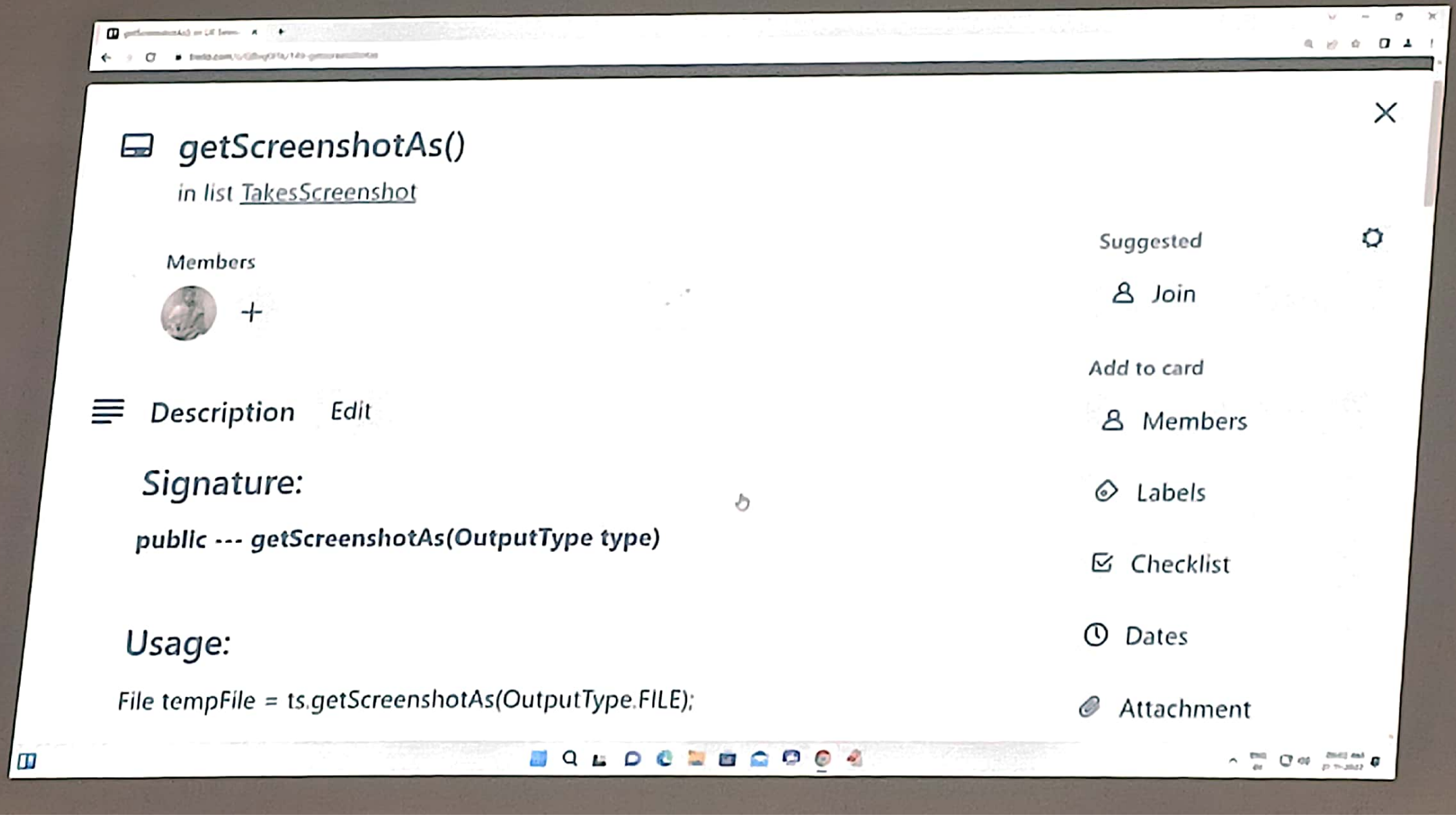
Steps to take Webpage Screenshot:

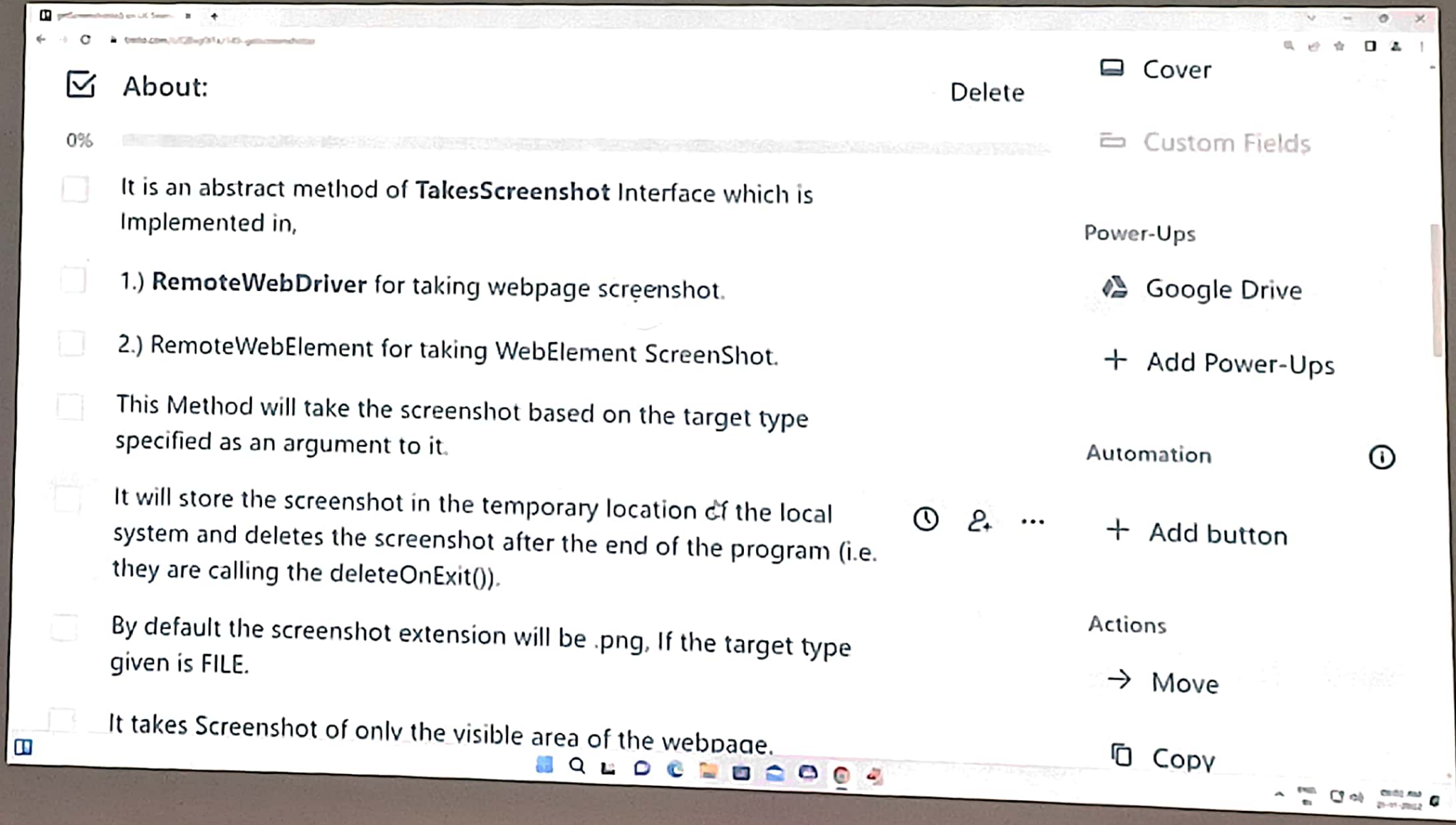
Delete

0%

- [ ] 1.) TypeCast the WebDriver object into TakesScreenshot Interface.
- [ ] 2.) Call the getScreenshotAs() with the target OutputType as FILE.
- [ ] 3.) Store the temporary ScreenShot returned by getScreenshotAs() in the File class reference variable.
- [ ] 4.) Create the destination file, File class object with the destination file path by calling the **File(String pathName)** constructor.
- [ ] 5.) Copy the temporary **File** class object given by getScreenshotAs() into the destination **File** class object using **copyFile(File srcFile, File destFile)** Static method of **FileUtils** Class from **commons-io** library.

Add an item


getScreenshotAs()
in list TakesScreenshot
Members
Suggested
Join
Add to card
Members
Labels
Checklist
Dates
Attachment
Description
Edit
Signature:
public --- getScreenshotAs(OutputType type)
Usage:
File tempFile = ts.getScreenshotAs(OutputType.FILE);


About:
Delete
0%
It is an abstract method of **TakesScreenshot** Interface which is Implemented in,
1.) **RemoteWebDriver** for taking webpage screenshot.
2.) RemoteWebElement for taking WebElement ScreenShot.
This Method will take the screenshot based on the target type specified as an argument to it.
It will store the screenshot in the temporary location of the local system and deletes the screenshot after the end of the program (i.e. they are calling the deleteOnExit()).
By default the screenshot extension will be .png, If the target type given is FILE.
It takes Screenshot of only the visible area of the webpage.
Cover
Custom Fields
Power-Ups
Google Drive
Add Power-Ups
Automation
Add button
Actions
Move
Copy

Implemented in,

1.) **RemoteWebDriver** for taking webpage screenshot.

2.) RemoteWebElement for taking WebElement ScreenShot.

This Method will take the screenshot based on the target type specified as an argument to it.

It will store the screenshot in the temporary location of the local system and deletes the screenshot after the end of the program (i.e. they are calling the deleteOnExit()).

By default the screenshot extension will be .png. If the target type given is FILE.

It takes Screenshot of only the visible area of the webpage.

Add an item

Power-Ups

Google Drive

+ Add Power-Ups

Automation

+ Add button

Actions

Move

Copy

Make template

NOTE:

Delete

Archive

Share

0%

Here the return type depends on the argument.

For Example: If the argument is FILE then the return type is File class object.

****It will take the current browser window screenshot.

Always remember that, It will shift the focus on the target window automatically and then takes the screenshot.

Add an item

OutputType:

Delete

0%

OutputType:

Delete

0%

OutputType is an interface which contains 3 static and final variables which is used to specify the target screenshot type for getScreenshotAs(). The final variables are:

1.) FILE - Datatype of this final variable is File.

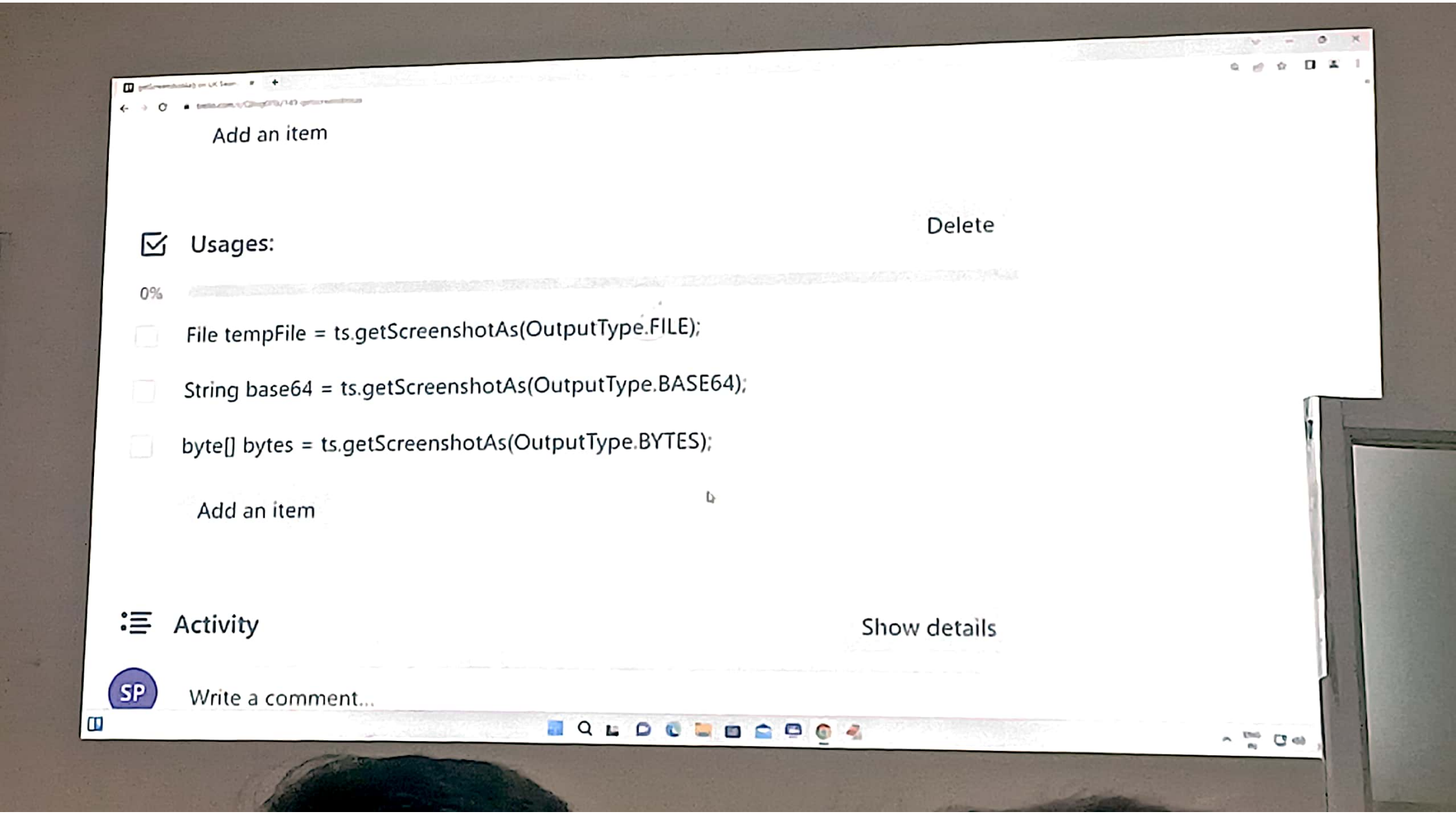
2.) BASE64 - Datatype of this final variable is String.

3.) BYTES - this final variable will give byte[] in terms of byte.

Add an item

Usages:

Delete


Add an item
Delete
Usages:
0%
File tempFile = ts.getScreenshotAs(OutputType.FILE);
String base64 = ts.getScreenshotAs(OutputType.BASE64);
byte[] bytes = ts.getScreenshotAs(OutputType.BYTES);
Add an item
Activity
Show details
SP
Write a comment...