Robot Framework:

* Robot Framework is a test automation framework for acceptance testing.
* It is based on Acceptance test-driven development (ATDD).
* Utilizes keyword-driven testing. We just use keywords to perform automation instead of using methods, classes etc.
* Provides test libraries implemented in Python or Java.
* Users can create new higher-level keywords from existing ones using the same syntax that is used for creating test cases.

A screenshot of a computer

Description automatically generated

* When we write our script using Robot Framework using Keywords, internally, those keyword statements execute the related WebDriver methods from Selenium and other libraries. These methods send HTTP requests to the WebDriver through JSON wire protocol/W3C protocol (Selenium 4) to perform the automation on the web browser through API calls.
* To write Robot framework scripts in PyCharm, we need to install Intellibot plugin.

A screenshot of a computer program

Description automatically generated

Pre-requisites:

1. Install the required packages:

* ‘pip install robotframework’ and ‘pip show robotframework’ to install and verify Robot framework.
* ‘pip install --upgrade robotframework’ to upgrade the to the latest robot framework package.
* ‘pip list’ to see all the installed packages.
* ‘pip install robotframework-seleniumlibrary’ to install selenium library for robot framework.
* ‘pip uninstall <package>’ to uninstall any package.

1. Add the above packages to the Project in PyCharm:

* File > Settings > Project > Python Interpreter.
* Check is the packages are installed, if not add them using ‘+’.
* Restart PyCharm.

1. Install Intellibot Plugin to PyCharm:

* File > Settings > Project > Plugins.
* Search in Marketplace for ‘Intellibot@SeleniumLibrary Patched’ and install it.
* Restart PyCharm.

Refer to this link for documentation about Robot Framework Keywords: <https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html>

Data-driven Testing suing Excel and CSV Pre-requisites:

1. Install the ‘robotframework-datadriver’ library:

File > Settings > Project > Python Interpreter> Add package or

‘pip install robotframework-datadriver’

Database Testing:

1. For performing DB Testing using Robot Framework, we must install two libraries:
2. ‘robotframework-databaselibrary’ library
3. ‘pymysql’ Library for MySQL DB libraries.

How to run test suites sequentially:

1. Specify the folder containing the test case files and run from terminal:

‘robot <folder\_path>\’

1. We can also use regular expressions to run only files that match it:

‘robot <folder\_path>\<filenames\_regex>.robot’

Eg: robot Testcases\Lo\*.robot executes all the files that start with ‘Lo’

robot Testcases\\*.robot executes all the files in the folder.

Parallel Testing Pre-reqs.:

1. Install ‘robotframework-pabot’ library.
2. To run 2 tests parallely:

‘pabot –processes 2 <folder>\<filename\_regex>.robot’

1. To add an output directory for results:

‘pabot –processes 2 –outputdir <outputfolder> <folder>\<filename\_regex>.robot’

What are bat files and how to run from a bat file?

.bat or Batch files are script files that store commands to be executed in a serial order. It helps automate routine tasks without requiring user input or intervention. Some common applications of batch files include loading programs, running multiple processes, or performing repetitive actions in a sequence in the system.

1. Create a ‘run.bat’ file, add the path to the Project folder.
2. Add the command to run the files:

‘robot <folder\_path>\<file\_name>.robot’ or ‘pabot ……’

1. Run the .bat file in cmd prompt.

What are headless browsers and how to run tests on them?

Headless means a Web Browser without a User Interface. Headless browser testing increases the efficiency of testing your web applications and provides ease of testing those apps. Therefore, we can’t see anything when you use a headless browser to access any website. However, the program runs in the background.

To run tests on headless chrome, Change the Browser from ‘Chrome’ to ‘headlesschrome’.

Advantages of using Headless browsers:

1. Resource Efficiency: Headless browsers don’t require the resources to render and display web content, such as graphical elements and animations. This makes them suitable for environments with limited resources.

2. Speed: Without rendering and displaying content, headless browsers can often load and interact with web pages more quickly than their GUI counterparts.

3. Scalability: They are easier to scale since they can run in the background without consuming graphical resources.

4. Automation: Ideal for automating interactions with web pages, such as filling out forms, clicking buttons, and navigating through pages.

Integrate Robot with Jenkins:

1. Download the ‘Jenkins.war’ file.

2. Create a folder and place it in that.

3. Run cmd prompt from that folder where the ‘.war’ file is located.

4. Run: ‘java -jar jenkins war –httpPort:<PortNumber>’ This will launch Jenkins web service container on that port on localhost.

5. Go to Jenkins and create a new item. Choose ‘Freestyle Project’, give a name and configure it:

In ‘Build Steps’ mention the .bat file it is supposed to run:

‘cd <path\_to\_bat\_file>’

‘run.bat’