YAKSHA HEALTH APP WITH TYPESCRIPT AND PLAYWRIGHT - PL2

Mymedic automation using playwright

Usecase summary

Project Name: healthapp.yaksha app – Medical Record Management System
Use Case Summary: healthapp.yaksha is a healthcare application designed to manage Electronic Medical Records (EMR). it allows users to view, search, and manage patient records. It features functionality such as adding/editing patient records, filtering data by doctor and department, and exporting records. The primary use case is to automate the process of medical record management, ensuring efficient and reliable operations for healthcare providers.
Technology Stack:

- **Automation Tool:** Playwright (for testing)

Key Features:

- **Patient Record Management:** Add, edit, and delete patient records.
- **Filtering and Search:** Search medical records by date range, doctor, department, and more.
- **Export Functionality:** Export records for offline access.

Expected Outcomes:

- Automate key healthcare operations like patient record handling, filtering, and validation.
- Ensure the accurate retrieval and modification of medical records, enhancing operational efficiency.

Overview of the application

Pages/Features that are to be focused for the application

Please use the Application URL https://healthapp.yaksha.com

PROBLEM STATEMENT

Need to automate the following activities using playwright+typescript

You will be given a excel file named to validate and search data

Path	**File**	**Description**
src\data	result .xlsx	1. Contains data to read from excel file.
src\tests\commonMethods	readExcel(filePath, sheetName)	Should be implemented such way that it reads the data from shared “sheetName” from excel of given “filePath” and return the data in Record<string, string>

Mymedic automation using playwright

src\ pages	<ul><li>AppointmentPage</li><li>UtilitiesPage</li><li>DispensaryPage</li><li>ProcurementPage</li><li>LoginPage</li><li>PatientPage</li><li>ADTPage</li><li>RadiologyPage</li><li>LaboratoryPage</li></ul>	1. All core activities to be performed here. 2. The comments associated with each templated method here describe the expectation. 3. Declare any variable/object you need to share data/status between different methods. 4. Do not modify the signature of methods declared here.
src\tests	keywords.ts	Implement methods SearchPatients() and VerifyResults()

Here's a detailed table format for the test cases to be tested

Test Case No.	Test Case Name	Test Steps to be performed	Path & Method Used	Expected Result
1	Verify Login with Valid Credentials	1.the application will read the result .xlsx file to fetch the user name and password using "login" string from common methods 2.it should call the method Use performLogin () 3.perform login method will perform authentication with the username and password 4. Verify admin name is visible on the home page.	**Reference path** \src\pages\LoginPage **methods** performLogin()	Successfully logs in with provided credentials. The user is logged in the admin page Returns true for success; otherwise, false.
2	Verify Page Navigation and Load Time for Billing Counter	1. Use verifyBillingCounterLoadState() to check module load. 2. Open "Change Billing Counter" module using ChangeBillingCounter. 3. Set acceptable load time: 1000ms. 4. Verify counter presence; select the first counter if available. 5. Log a message if no counters are found.	**Reference path** \src\ pages\ **UtilitiesPage** **methods** verifyBillingCounterLoadState()	Navigates to "Change Billing Counter". Proceeds if counters are available and loaded within 1 second; logs a message otherwise.
3	Activate Counter in Dispensary	1. Use verifyActiveCounterMessageInDispensary() to: - Navigate to the Dispensary page. - Select a random counter if available. - Activate the counter and verify the activation message. 2. Log counter selection and activation status.	**Reference path** \src\ pages\ **DispensaryPage** **methods** verifyActiveCounterMessageInDispensary()	Counter activation message matches the selected counter name. Returns true for success; false otherwise.
4	Purchase Request List Load	1.Use verifyPurchaseRequestListElements() to check verify of elements: purchaseRequest, purchaseOrder, goodsArrivalNotification, quotations, settings, reports,	**Reference path** \src\ pages\ **ProcurementPage** **methods** verifyPurchaseRequestListElements()	All elements are present and visible on the procurement page Returns true for success; logs missing elements and returns false otherwise.

Test Case No.	Test Case Name	Test Steps to be performed	Path & Method Used	Expected Result
		favoriteButton, okButton, printButton, firstButton, previousButton, nextButton, lastButton.		
5	Verify Error Message While Adding New Lab Test	1. Navigate to Laboratory > Settings. 2. Select "Add New Lab Test". 3. Click "Add" without providing inputs. 4. Capture and verify the error message: "Lab Test Code Required".	**Reference path** \src\ pages\ **LaboratoryPage** **methods** verifyErrorMessage()	Error message: "Lab Test Code Required" is displayed. Logs success or failure and ensures the modal is closed.
6	Handle Alert on Radiology Module	1. Navigate to Radiology module and select "List Request" sub-module. 2. Apply filter using dates . 3. the application will read the data from excel file using the common methods name (“DateRange”) (eg fromDate: 01-01-2020 to toDate: 11-11-2024.) 3. Click "Add Report" button. 4. Use handleAlert() to verify and accept the alert if the message matches. 5. the application will check for the matched data .	**Reference path** \src\ pages\ **RadiologyPage** **methods** handleAlert()	Alert dialog matches expected message and is accepted. Returns true for success; false for failure in handling the alert.
7	Data-Driven Testing for Patient Search	1. Navigate to Patient Section. 2. application will use the common method to use read the excel to search for patient which is there in the dataset using the name (“patientnames”) 3. Use the search bar to find patients. 4. Compare retrieved names with expected names from Excel. 5. Log success or failure for each match.	**Reference path** \src\ pages\ **PatientPage** **methods** searchAndVerifyPatients()	Matches all patient names from Excel. Returns true for success; logs mismatches and returns false if any fail.
8	Error Handling and Logging in Purchase Request List	1. Navigate to Procurement module. 2. Apply invalid date filter. 3. Click "OK". 4. Capture and verify the error message: "Date is not between range. Please enter again."	**Reference path** \src\ pages\ **ProcurementPage** **methods** verifyNoticeMessageAfterIncorrectFilters()	Triggers and verifies the error message for invalid date range. Logs success for match; failure for mismatch.
9	Verify Assertion for Counter Activation	1. Navigate to the Dispensary section. 2. Verify the visibility of the counter button. 3. Activate counter using activateCounter button. 4. Verify deactivateCounterButton visibility.	**Reference path** \src\ pages\ **DispensaryPage** **methods** verifyCounterisActivated()	Returns true if counter activation is successful. Returns false if activation fails or required element is not found.
10	Verify Locator Strategy for Appointment Search	1. Navigate to Appointment page. 2. Verify visibility of patient list. 3. Use the search bar to find a patient by name or hospital code. 4. Verify search results.	**Reference path** \src\ pages\ **AppointmentPage** **methods**	Verify that each patient's name in the result matches the search term

Test Case No.	Test Case Name	Test Steps to be performed	Path & Method Used	Expected Result
			searchAndVerifyPatientList()	
11	Assertions are defined for required fields (e.g., Username, Password) and optional fields (e.g., Remember Me).	1. Verify presence of username, password, and "Remember Me" checkbox on the login page. 2. If already logged in, log out to reset the state. 3. Use Excel data to perform login. 4. Return true if login is successful; otherwise, false.	**Reference path** **\src\pages\LoginPage** **Methods**: verifyThePresenceOfLoginFields()	Returns true if the user successfully logs in after verifying the presence of the login fields, otherwise false
12	Switching Between Pages & Windows	1. Navigate to the Claim Management module and open the Insurance Provider screen. 2. Validate the URL to ensure the correct screen is loaded. 3. Go to the Dashboard module. 4. Return to the Insurance Provider screen using browser navigation and revalidate the URL.	**Reference path** **\src\pages\claimManagementPage** **Methods**: verifyWindowNavigation()	Navigation is successful, and URL validation confirms the correct Insurance Provider screen is loaded upon return.
13	Locator Strategies with XPath & CSS	1. Read data for date range using CommonMethods.readExcel(filePath, "DateRange"). 2. Verify search functionality using dispensaryPage.verifySearchFunctionality(data).	**Reference path** **\src\ pages\ DispensaryPage** **Methods**: verifySearchFunctionality()	A record containing the "FromDate" for filtering the User Collection Report
14	Manage Web Element Interactions	1. Verify bill details using billingPage.verifyBillDetails(). 2. Navigate to the Billing module and select a counter. 3. Access "Return Bills," filter by fiscal year and invoice number, and initiate the search. 4. Validate the search results contain at least one entry.	**Reference path** **\src\ pages\ billingPage** **Methods**: verifyBillDetails()	throws Logs an error message if there is an issue during the verification process.
15	Implement Wait Strategies	1. Navigates to the Claim Management module, selects an insurance provider, and opens the Bill Review section. 2. Using the excel data, filters the bills using a "From Date" and loads the results.	**Reference path** **\src\ pages** **claimManagementPage** **Methods**: verifyBillReview()	returns true if the results are successfully loaded and verified; false otherwise

Test Case No.	Test Case Name	Test Steps to be performed	Path & Method Used	Expected Result
		3. Verifies that the results table contains entries.		
16	Verify Add/Remove Departments	1. Navigate to the Settings module and add a new department with a unique code and name. 2. Fill in the fields and submit the form to save the department. 3. Search for the created department and update its status to inactive by selecting 'No.' 4. Verify the success message is displayed after updating.	**Reference path** **\src\pages\SettingsPage** **Methods**: verifyAddAndEditDepartment()	A new department is added, its status updated to inactive, and a success message is displayed; errors are logged if the process fails.

Learners will gain experience in building strongly-typed applications using React.js and managing data flow with **TypeScript**. They'll learn how to define interfaces, use types for error prevention, and improve code maintainability.

With **Playwright**, learners will learn to write and execute automated tests for the https://healthapp.yaksha.com

app. Key skills include:

- **Browser Automation**: Interacting with web elements and testing multiple browsers.
- **Assertions & Validations**: Ensuring app behavior meets expected results.
- **End-to-End Testing**: Automating real user interactions and validating overall app functionality.

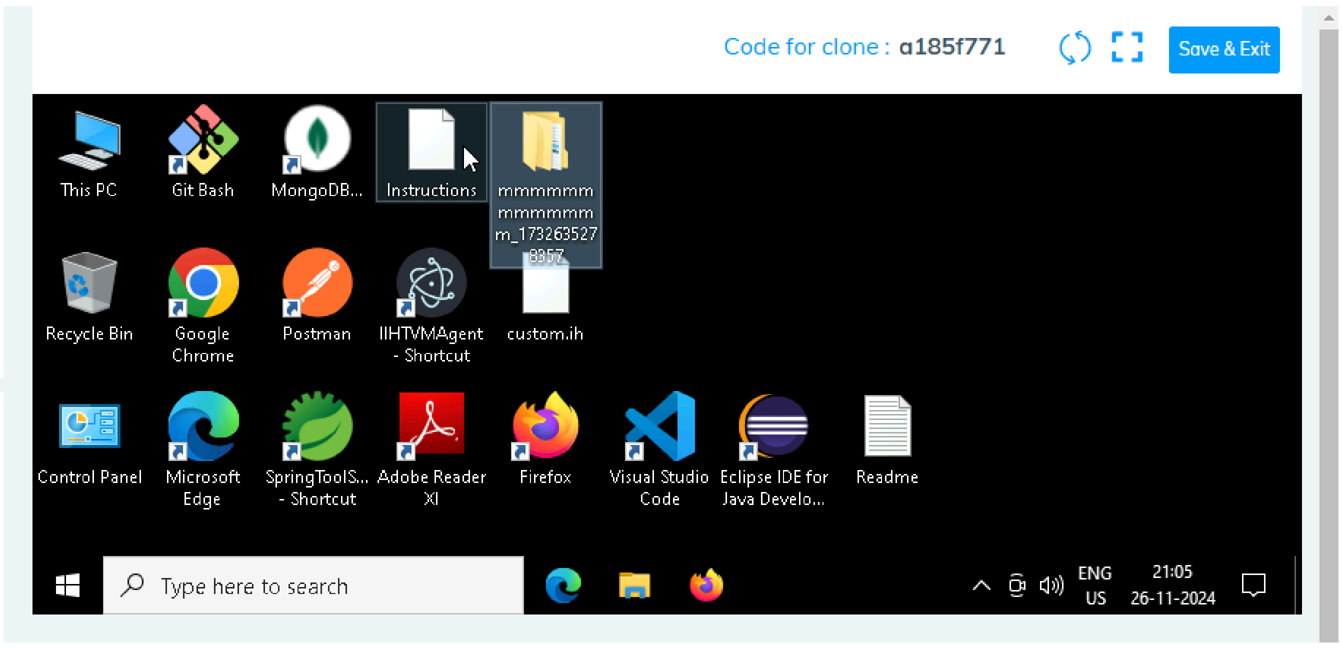
IMPLEMENTATION/FUNCTIONAL REQUIREMENT

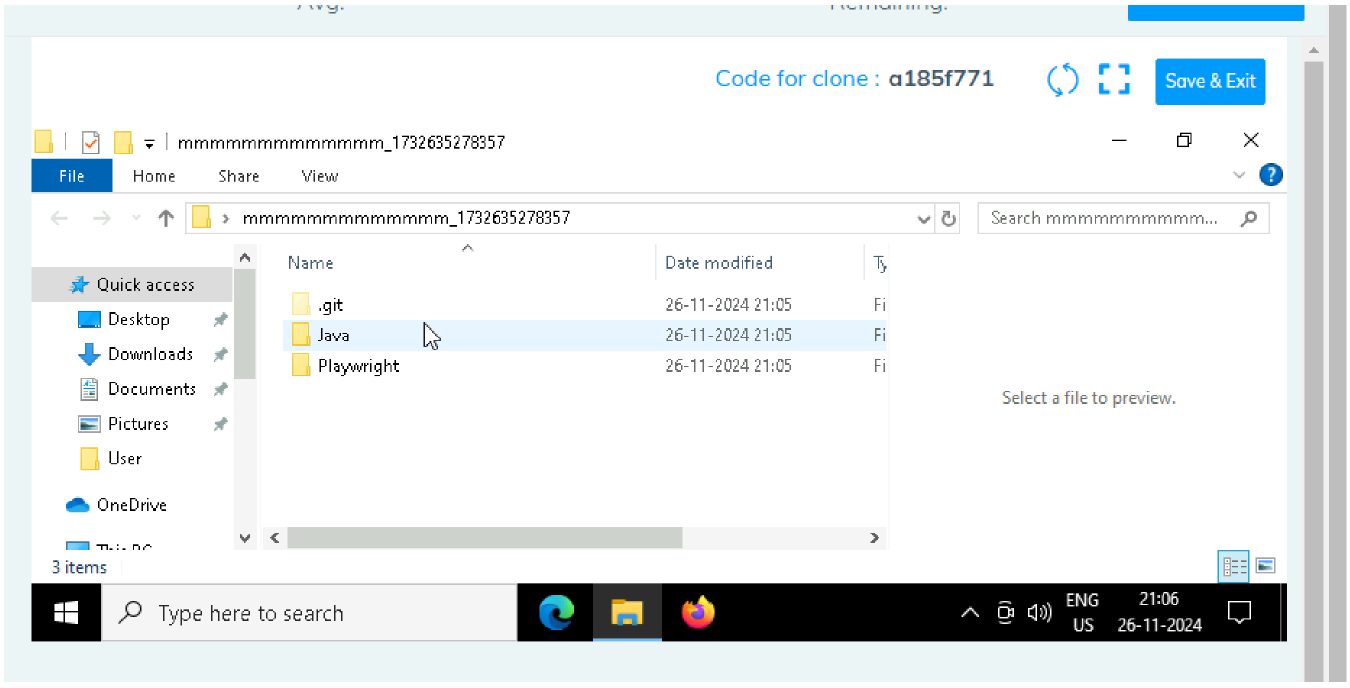
1.1 CODE QUALITY/OPTIMIZATIONS

1. Associates should have written clean code that is readable.
2. Associates need to follow SOLID programming principles.

Execution Steps:
Steps for Execution:

1. **Please open the folder created on the desktop with the email name you used to login.**

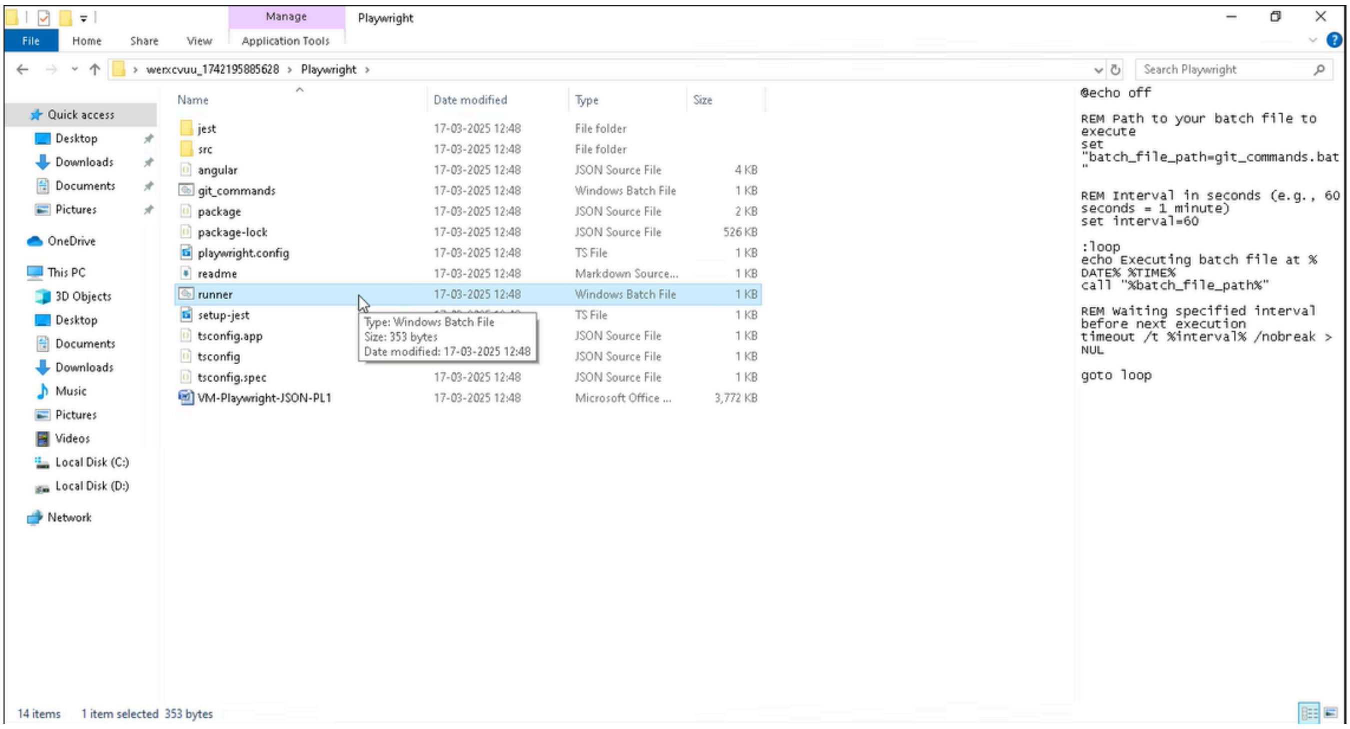




Name	Date modified	Ty
.git	26-11-2024 21:05	Fi
Java	26-11-2024 21:05	Fi
Playwright	26-11-2024 21:05	Fi

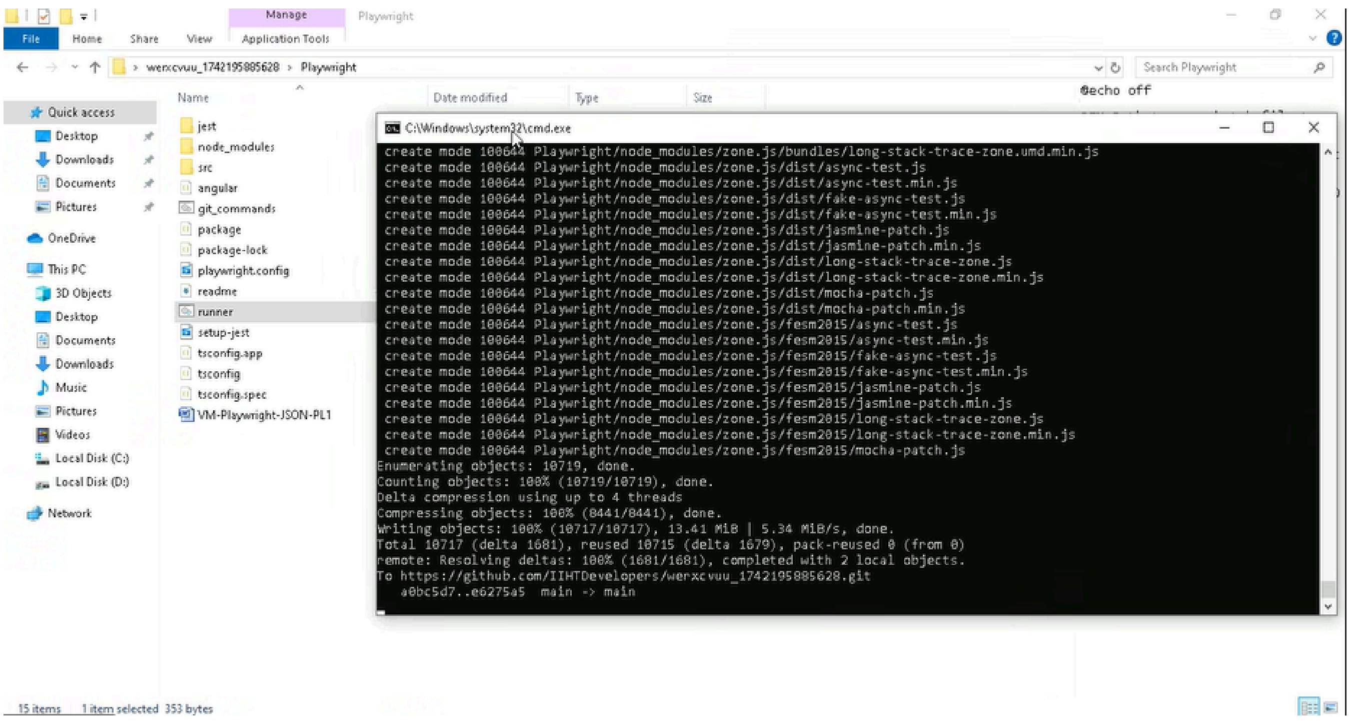
Mymedic automation using playwright

2. **Go into the Playwright folder and execute this “runner” file. This will keep pushing the code at regular intervals.**



Name	Date modified	Type	Size
jest	17-03-2025 12:48	File folder	
src	17-03-2025 12:48	File folder	
angular	17-03-2025 12:48	JSON Source File	4 KB
git_commands	17-03-2025 12:48	Windows Batch File	1 KB
package	17-03-2025 12:48	JSON Source File	2 KB
package-lock	17-03-2025 12:48	JSON Source File	526 KB
playwright.config	17-03-2025 12:48	TS File	1 KB
readme	17-03-2025 12:48	Markdown Source...	1 KB
runner	17-03-2025 12:48	Windows Batch File	1 KB
setup-jest		TS File	1 KB
tsconfig.app		JSON Source File	1 KB
tsconfig		JSON Source File	1 KB
tsconfig.spec	17-03-2025 12:48	JSON Source File	1 KB
VM-Playwright-JSON-PL1	17-03-2025 12:48	Microsoft Office ...	3,772 KB

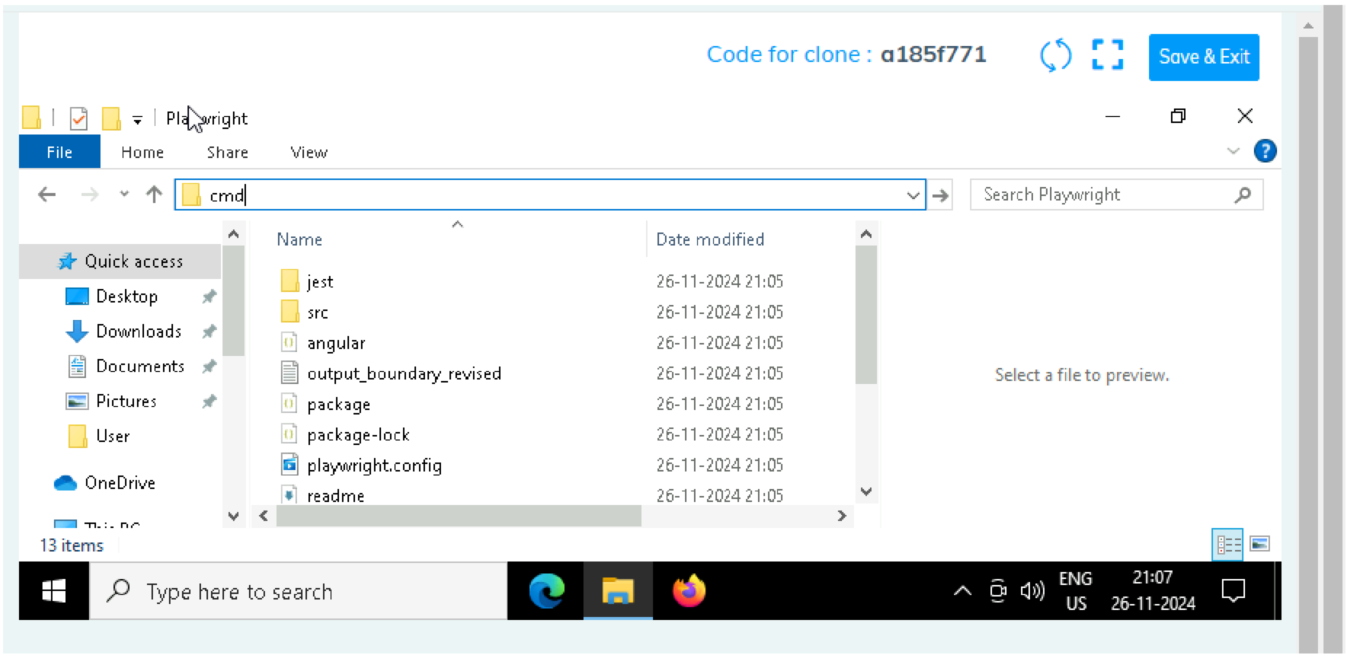
```
@echo off

REM Path to your batch file to
execute
set
"batch_file_path=git_commands.bat
"

REM Interval in seconds (e.g., 60
seconds = 1 minute)
set interval=60

:loop
echo Executing batch file at %
DATE% %TIME%
call "%batch_file_path%"

REM Waiting specified interval
before next execution
timeout /t %interval% /nobreak >
NUL

goto loop
```



```
 create mode 100644 Playwright/node_modules/zone.js/bundles/long-stack-trace-zone.umd.min.js
 create mode 100644 Playwright/node_modules/zone.js/dist/async-test.js
 create mode 100644 Playwright/node_modules/zone.js/dist/async-test.min.js
 create mode 100644 Playwright/node_modules/zone.js/dist/fake-async-test.js
 create mode 100644 Playwright/node_modules/zone.js/dist/fake-async-test.min.js
 create mode 100644 Playwright/node_modules/zone.js/dist/jasmine-patch.js
 create mode 100644 Playwright/node_modules/zone.js/dist/jasmine-patch.min.js
 create mode 100644 Playwright/node_modules/zone.js/dist/long-stack-trace-zone.js
 create mode 100644 Playwright/node_modules/zone.js/dist/long-stack-trace-zone.min.js
 create mode 100644 Playwright/node_modules/zone.js/dist/mocha-patch.js
 create mode 100644 Playwright/node_modules/zone.js/dist/mocha-patch.min.js
 create mode 100644 Playwright/node_modules/zone.js/fesm2015/async-test.js
 create mode 100644 Playwright/node_modules/zone.js/fesm2015/async-test.min.js
 create mode 100644 Playwright/node_modules/zone.js/fesm2015/fake-async-test.js
 create mode 100644 Playwright/node_modules/zone.js/fesm2015/fake-async-test.min.js
 create mode 100644 Playwright/node_modules/zone.js/fesm2015/jasmine-patch.js
 create mode 100644 Playwright/node_modules/zone.js/fesm2015/jasmine-patch.min.js
 create mode 100644 Playwright/node_modules/zone.js/fesm2015/long-stack-trace-zone.js
 create mode 100644 Playwright/node_modules/zone.js/fesm2015/long-stack-trace-zone.min.js
 create mode 100644 Playwright/node_modules/zone.js/fesm2015/mocha-patch.js
Enumerating objects: 10719, done.
Counting objects: 100% (10719/10719), done.
Delta compression using up to 4 threads
Compressing objects: 100% (8441/8441), done.
Writing objects: 100% (10717/10717), 13.41 MiB | 5.34 MiB/s, done.
Total 10717 (delta 1681), reused 10715 (delta 1679), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1681/1681), completed with 2 local objects.
To https://github.com/IIHTDevelopers/werxcvuu_1742195885628.git
   a0bc5d7..e6275a5  main -> main
```

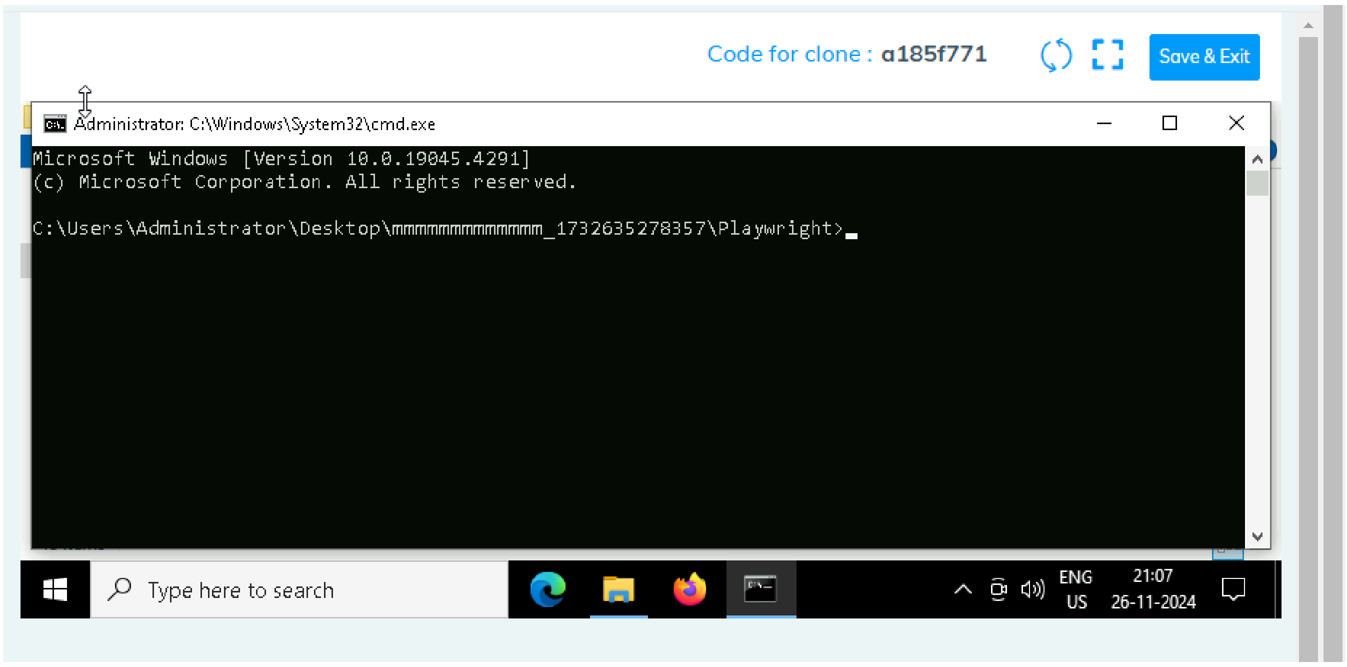
3. **Open command prompt with its location and use below command:**

   code .

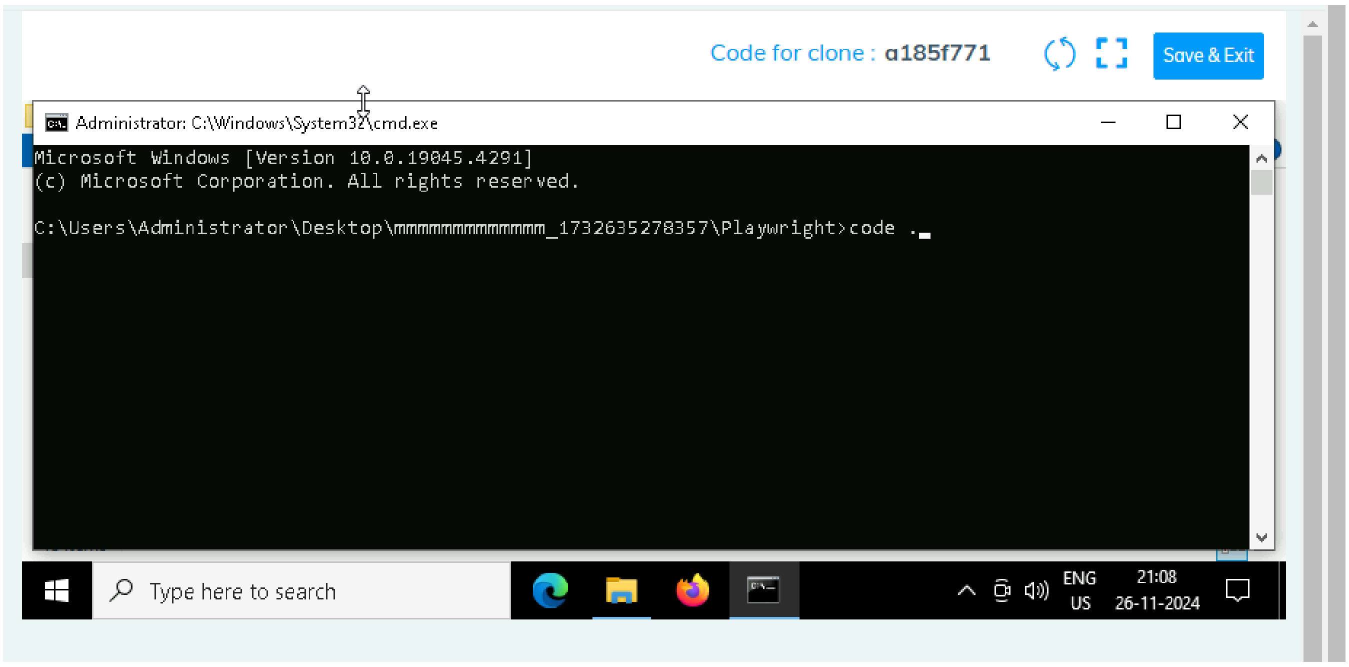


```
cmd
```

Name	Date modified
jest	26-11-2024 21:05
src	26-11-2024 21:05
angular	26-11-2024 21:05
output_boundary_revised	26-11-2024 21:05
package	26-11-2024 21:05
package-lock	26-11-2024 21:05
playwright.config	26-11-2024 21:05
readme	26-11-2024 21:05


Code for clone : a185f771
Save & Exit
Administrator: C:\Windows\System32\cmd.exe
Type here to search
ENG US
21:07 26-11-2024

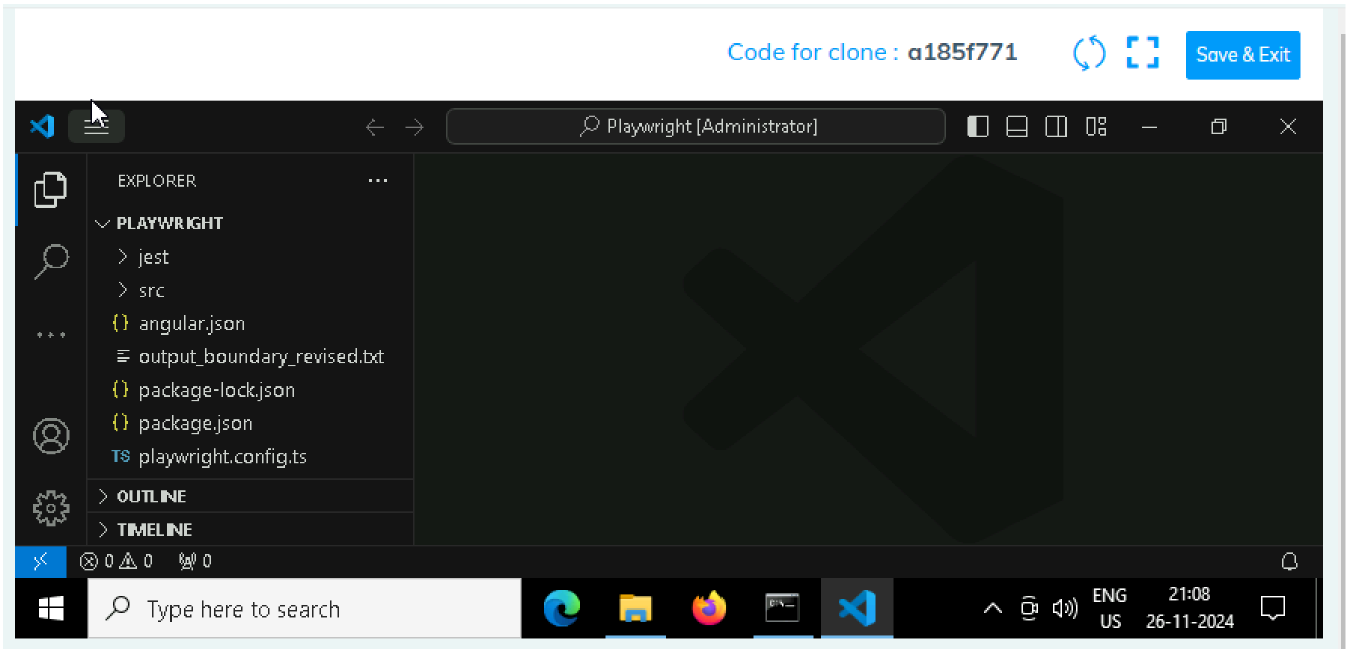
```
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Desktop\mmmmmmmmmmmmm_1732635278357\Playwright>
```


Code for clone : a185f771
Save & Exit
Administrator: C:\Windows\System32\cmd.exe
Type here to search
ENG
US
21:08
26-11-2024

```
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Desktop\mmmmmmmmmmmmm_1732635278357\Playwright>code .
```

Mymedic automation using playwright

4. **Once VsCode is open. Please open it's terminal by:**



5. **Install all dependencies using**

   npm install

6. **Install playwright**

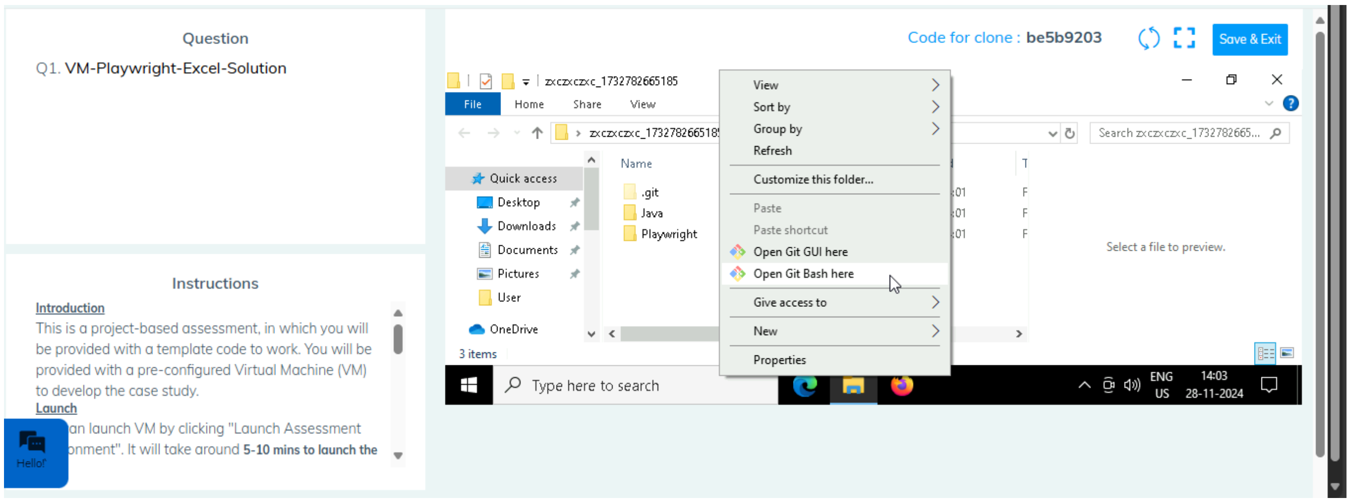
   npx playwright install

7. **Run the Tests**:

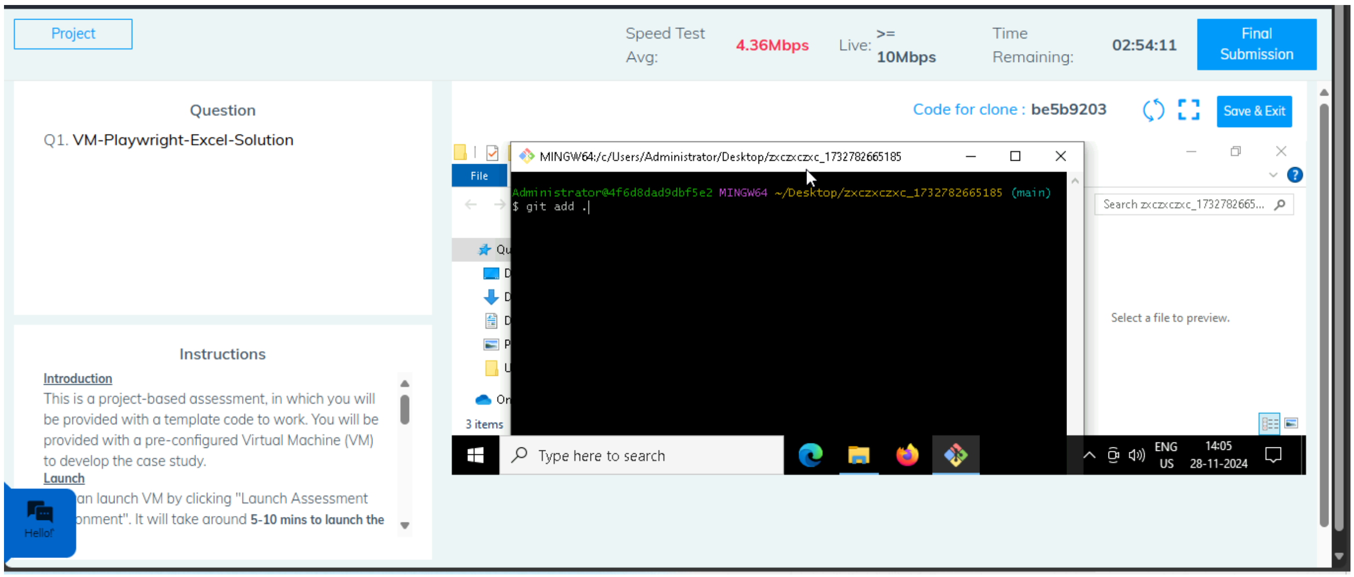
   npx playwright test ./src/tests/PL1_testcases/yaksha.spec.ts

8. **Once you have executed the test cases. Now it is necessary to push your code to git. For this, please go inside the folder created on desktop with the email id you have used to login and then:**

1. **Open gitbash**

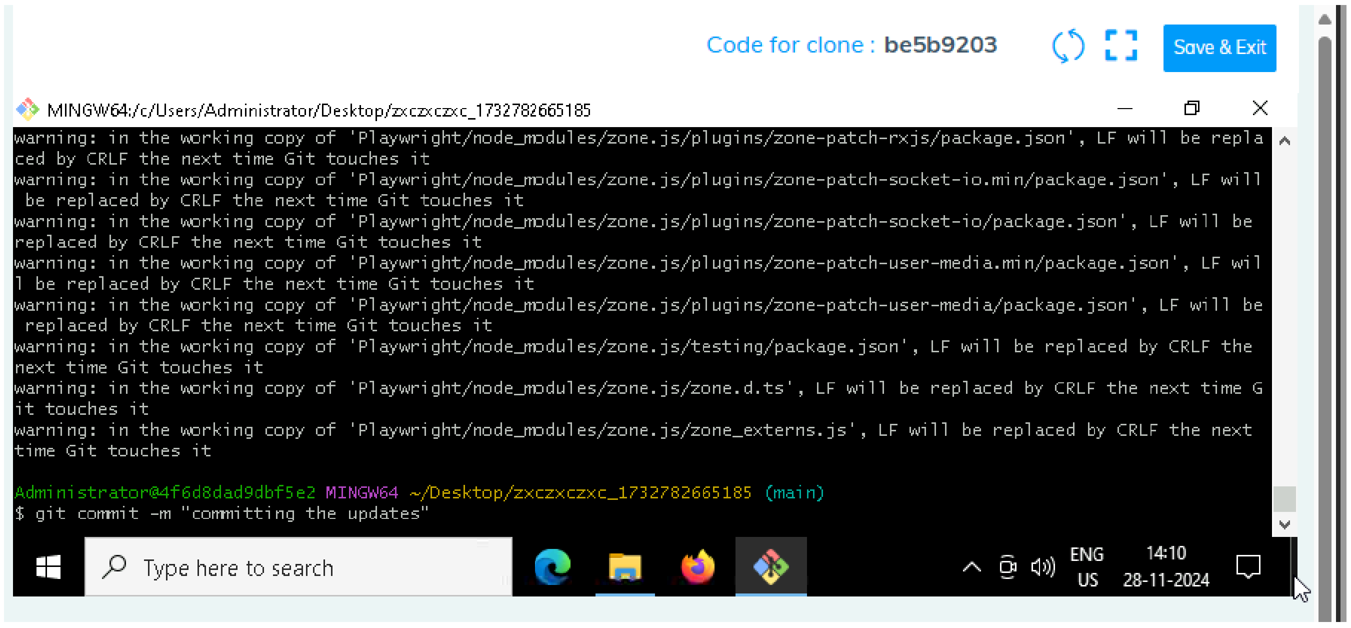

Question
Q1. VM-Playwright-Excel-Solution
Instructions
Introduction
This is a project-based assessment, in which you will be provided with a template code to work. You will be provided with a pre-configured Virtual Machine (VM) to develop the case study.
Launch
an launch VM by clicking "Launch Assessment onment". It will take around 5-10 mins to launch the
Hello!
Code for clone : be5b9203
Save & Exit
zxczxczxc_1732782665185
File
Home
Share
View
Quick access
Desktop
Downloads
Documents
Pictures
User
OneDrive
3 items
Name
.git
Java
Playwright
View
Sort by
Group by
Refresh
Customize this folder...
Paste
Paste shortcut
Open Git GUI here
Open Git Bash here
Give access to
New
Properties
Search zxczxczxc_1732782665...
Select a file to preview.
Type here to search
ENG US
14:03
28-11-2024

2. **Add all files**


Project
Speed Test Avg:
4.36Mbps
Live: >= 10Mbps
Time Remaining:
02:54:11
Final Submission
Question
Q1. VM-Playwright-Excel-Solution
Instructions
Introduction
This is a project-based assessment, in which you will be provided with a template code to work. You will be provided with a pre-configured Virtual Machine (VM) to develop the case study.
Launch
an launch VM by clicking "Launch Assessment onment". It will take around 5-10 mins to launch the
Hello!
Code for clone : be5b9203
Save & Exit
MINGW64:/c/Users/Administrator/Desktop/zxczxczxc_1732782665185
Search zxczxczxc_1732782665...
Select a file to preview.
3 items
Type here to search
ENG US
14:05
28-11-2024

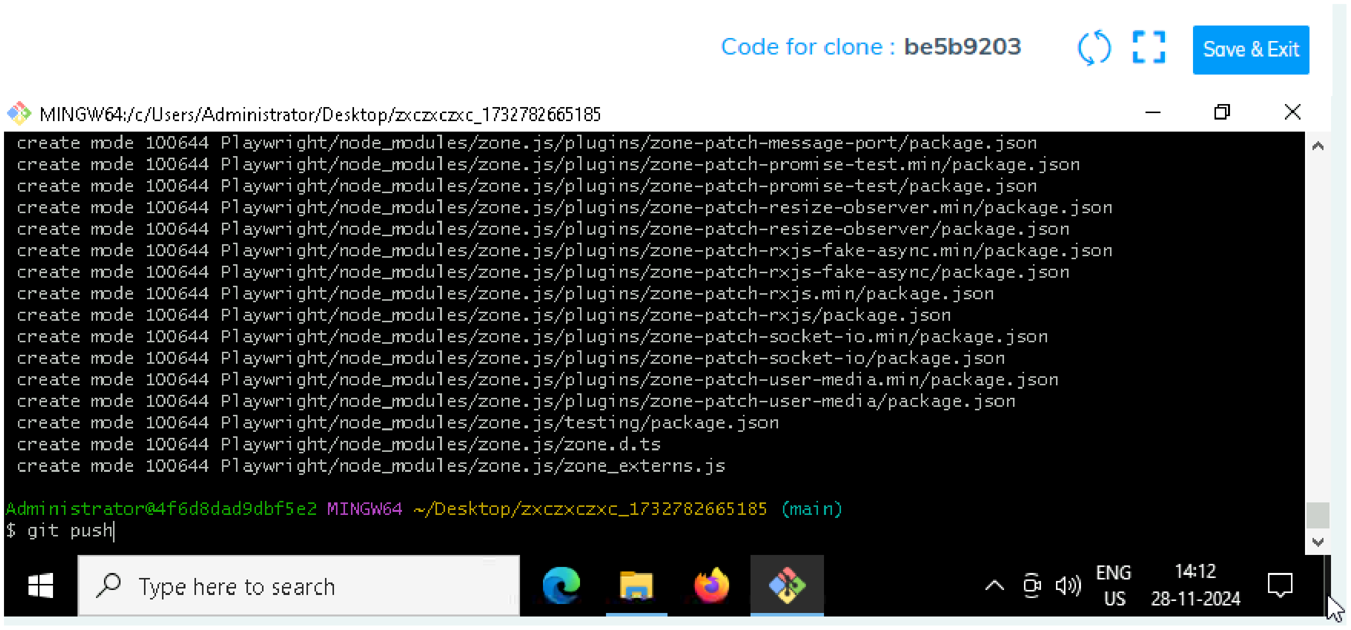
```
Administrator@4f6d8dad9dbf5e2 MINGW64 ~/Desktop/zxczxczxc_1732782665185 (main)
$ git add .
```

3. **Commit the changes**


Code for clone : be5b9203
Save & Exit
MINGW64:/c/Users/Administrator/Desktop/zxczxczxc_1732782665185

```
warning: in the working copy of 'Playwright/node_modules/zone.js/plugins/zone-patch-rxjs/package.json', LF will be repla
ced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/zone.js/plugins/zone-patch-socket-io.min/package.json', LF will
 be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/zone.js/plugins/zone-patch-socket-io/package.json', LF will be
replaced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/zone.js/plugins/zone-patch-user-media.min/package.json', LF wil
l be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/zone.js/plugins/zone-patch-user-media/package.json', LF will be
 replaced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/zone.js/testing/package.json', LF will be replaced by CRLF the
next time Git touches it
warning: in the working copy of 'Playwright/node_modules/zone.js/zone.d.ts', LF will be replaced by CRLF the next time G
it touches it
warning: in the working copy of 'Playwright/node_modules/zone.js/zone_externs.js', LF will be replaced by CRLF the next
time Git touches it

Administrator@4f6d8dad9dbf5e2 MINGW64 ~/Desktop/zxczxczxc_1732782665185 (main)
$ git commit -m "committing the updates"
```

4. **Push the changes**


Code for clone : be5b9203
Save & Exit
MINGW64:/c/Users/Administrator/Desktop/zxczxczxc_1732782665185

```
 create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-message-port/package.json
 create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-promise-test.min/package.json
 create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-promise-test/package.json
 create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-resize-observer.min/package.json
 create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-resize-observer/package.json
 create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-rxjs-fake-async.min/package.json
 create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-rxjs-fake-async/package.json
 create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-rxjs.min/package.json
 create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-rxjs/package.json
 create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-socket-io.min/package.json
 create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-socket-io/package.json
 create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-user-media.min/package.json
 create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-user-media/package.json
 create mode 100644 Playwright/node_modules/zone.js/testing/package.json
 create mode 100644 Playwright/node_modules/zone.js/zone.d.ts
 create mode 100644 Playwright/node_modules/zone.js/zone_externs.js

Administrator@4f6d8dad9dbf5e2 MINGW64 ~/Desktop/zxczxczxc_1732782665185 (main)
$ git push
```