YAKSHA HEALTH APP WITH TYPESCRIPT AND PLAYWRIGHT PL1-5

Use case summary

Project Name: https://yakshahrm.makemylabs.in/orangehrm-5.7 app – HR management system.

Use Case Summary: https://yakshahrm.makemylabs.in/orangehrm-5.7 is designed to manage Employee records. Records (HRMS). It allows users to view, search, and manage employee records. It features functionality such as adding/editing employee details, filtering data by department and role, and exporting records. The primary use case is to automate the process of employee data management, ensuring efficient and reliable operations for human resources teams in organizations

Technology Stack:

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

Key Features:

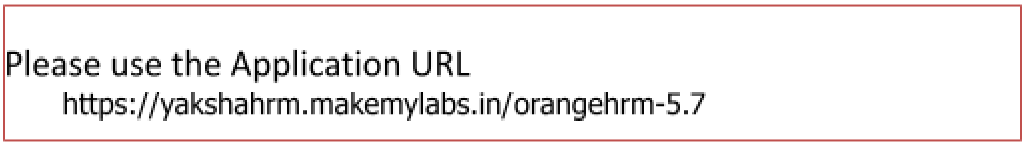
- **Patient Record Management:** Add, edit, and delete Employee records.
- **Filtering and Search:** Search Employee records by date range, name, department, and more.
- **Export Functionality:** Export records for offline access.

Expected Outcomes:

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

Overview of the application

Pages/Features that are to be focused for the application



Please use the Application URL
https://yakshahrm.makemylabs.in/orangehrm-5.7

PROBLEM STATEMENT

Need to automate the following activities using playwright+typescript

You will be given a few Json files in the Data folder login.json.

Path	File	Description
src\data	login.json	1. Contains data to read from a json file.

src\ pages	- AdminPage - BuzzPage - LoginPage - MyInfoPage	1. All core activities are 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.

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 Image could be Uploaded as profile Pic	1. The application should read the data from the login.json file and fetch the user’s name and password. 2. It should be called the method performLogin() and myinfo(). 3. Perform the myinfo method from MyInfoPage and perform the pic upload and set as profile pic. 4. Verify profile pic should be updated. **It is must implement this login functionality at first and then implement any other test case.**	**Reference path** \src\pages\MyInfoPage **methods** myInfo() **You can use the highlightElement method present in CommonMethods file to highlight the element before performing any action on it. It takes the locator as a parameter.**	image should get replaced
2	Ensure admin can access Job Categories	1. Click on admin page 2. Opens the Job Sub-tab 3. Select the job categories option	**Reference path** \src\ pages\AdminPage **methods** AdminEdit() **You can use the highlightElement method present in CommonMethods file to highlight the element before performing any action on it. It takes the locator as a parameter.**	Confirms that the Job Categories page is displayed, indicating successful navigation
3	Verify the Admin can sort the records	1.Login using valid credentials 2. Click on "Admin" tab in sidebar 3. Click on "sort" button	**Reference path** \src\ pages\AdminPage **methods** adminSortUsername()	List should get sorted accordingly

			You can use highlightElement method present in **CommonMethods file to highlight the element before performing any action on it. It takes the locator as a parameter.**	

4	Verify new Tab opens on clicking the "Upgrade" button	1. Navigates to the Admin section. 2. Click the upgrade button. 3. Waiting for a new tab to open. 4. Retrieves the URL of the newly opened tab.	**Reference path** \src\ pages\AdminPage **Methods:** upgrade() **You can use the highlightElement method present in CommonMethods file to highlight the element before performing any action on it. It takes the locator as a parameter.**	PicUpgrade to advance tab should open
5	Validate tooltip visibility on hovering over 'Help' button	1. Navigates to the Myinfo section 2. Hovers over the 'Help' button located in the top bar 3. Waits briefly to allow the tooltip to appear 4. Retrieves the tooltip text using the 'title' attribute 5. Passes the text for validation to confirm tooltip visibility and content	**Reference path** \src\ pages\MyinfoPage **methods** helpHover() **You can use the highlightElement method present in CommonMethods file to highlight the element before performing any action on it. It takes the locator as a parameter.**	Ensures that helpful contextual information is displayed when users hover over the help icon.
6	Confirm admin can access Module Configuration	6. Login using valid credentials 7. Click on "Admin" button in sidebar 8. Navigates to admin section 9. Opens the configuration tab 10. Select the module sub-tab	**Reference path** \src\pages\AdminPage **methods** : adminModules()	Confirms that the "Module Configuration" page is displayed
7	Validate photo sharing functionality with confirmation message	1. Navigates to the Buzz page 2. Click the "Share Photos" button 3. Enters a unique comment with a timestamp 4. Uploads an image file from local storage 5. Submits the post 6. Reloads the page to reflect the new post	**Reference path** \PageObjects\Pages\BuzzPage **methods** SharePhotoPost()	Ensures user can successfully share a photo post

8	Verify 'Most Commented Posts' tab selection	1. Navigates to the Buzz section 2. Click on Most Commented Post tab 3.. Verifies that the tab has active class applied	**Reference path** \src\pages\BuzzPage **methods** mostCommentTab()	Verifies that the tab has the active class applied

9	Ensure comment can be successfully added to a post	1. Navigates to the buzz section. 2. Clicks on the comment button for the first post 3. Fills in a unique comment using a timestamped message 4. Submits the comment by pressing Enter 5. Retrieves the most recent comment text 6. Collects all comment elements on the page 7. Verifies that the posted comment is present in the list	**Reference path** \src\ pages**BuzzPage** **methods** addCommentToPo st(). **You can use the highlightElement method present in CommonMethods file to highlight the element before performing any action on it. It takes locator as**	Ensures the comment submission flow is working correctly and the UI reflects the new comment as expected.
10	Verify post content can be edited successfully	1. Navigates to the Buzz section 2. Opens the options menu for the most recent post 3. Selects the edit option 4. Clears the existing post content 5. Fills in the new predefined post content (editPostText) 6. Saves the updated post 7. Verifies that the post now displays the updated content **Note :- If you cannot find the locator for the "Proceed" button, please try to keep the page at 75%-80%.**	**a parameter.** **Reference path** \src\pages\BuzzPage **methods** editPost() **You can use the highlightElement method present in CommonMethods file to highlight the element before performing any action on it. It takes the locator as a parameter.**	Confirms that users can update their shared posts and that those changes persist correctly.

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. URL https://yakshahrm.makemylabs.in/orangehrm-5.7

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.

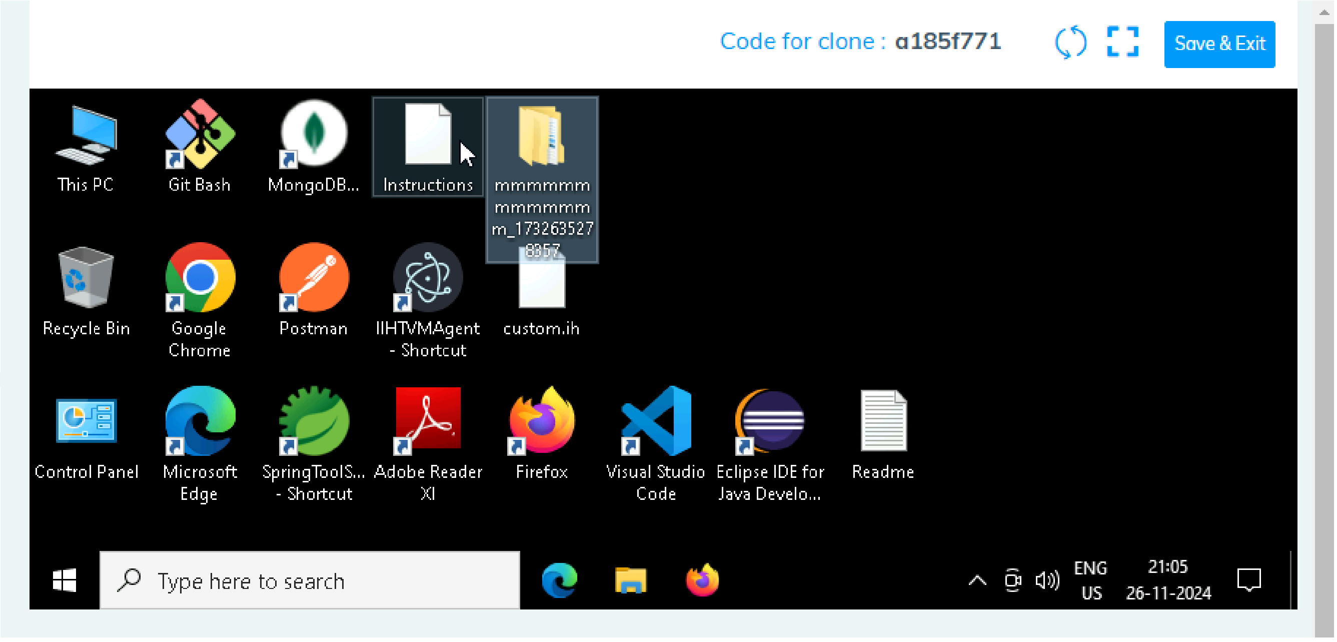
IMPLEMENTATION/FUNCTIONAL REQUIREMENT

1.1 Code Quality/Optimizations

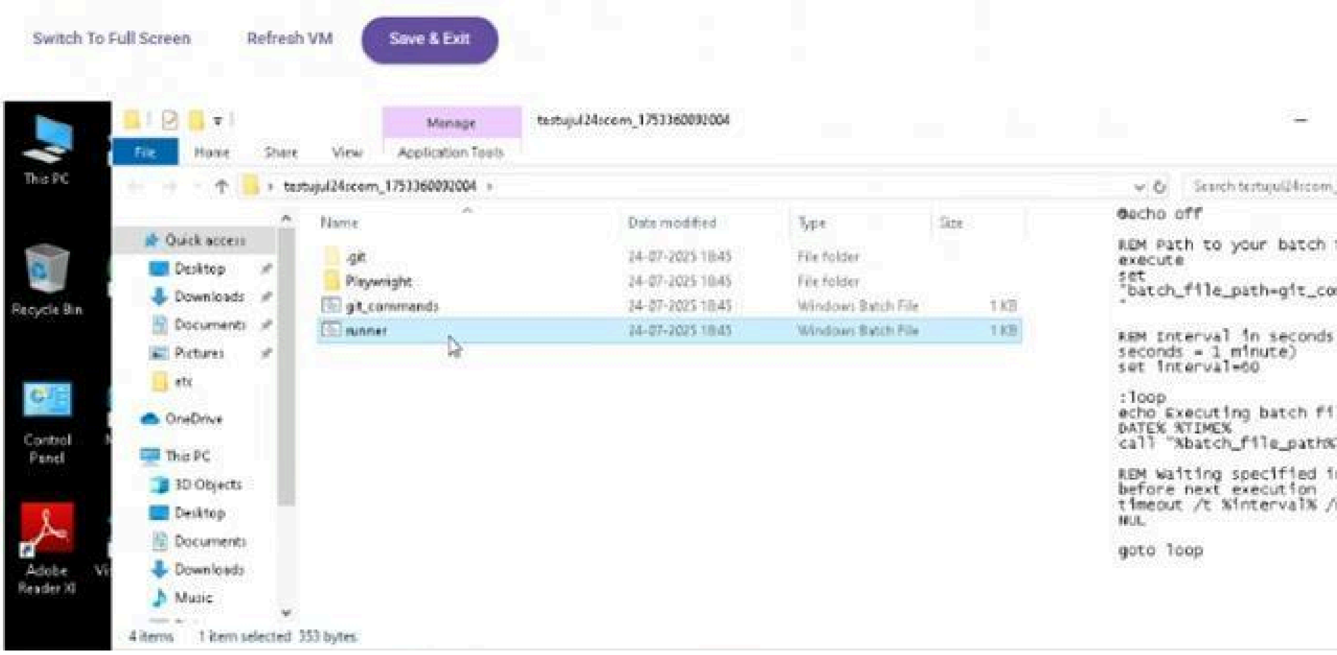
1. Associates should have written clean code that is readable.
2. Associates need to follow SOLID programming principles.

Execution Steps:
Steps for Execution:

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



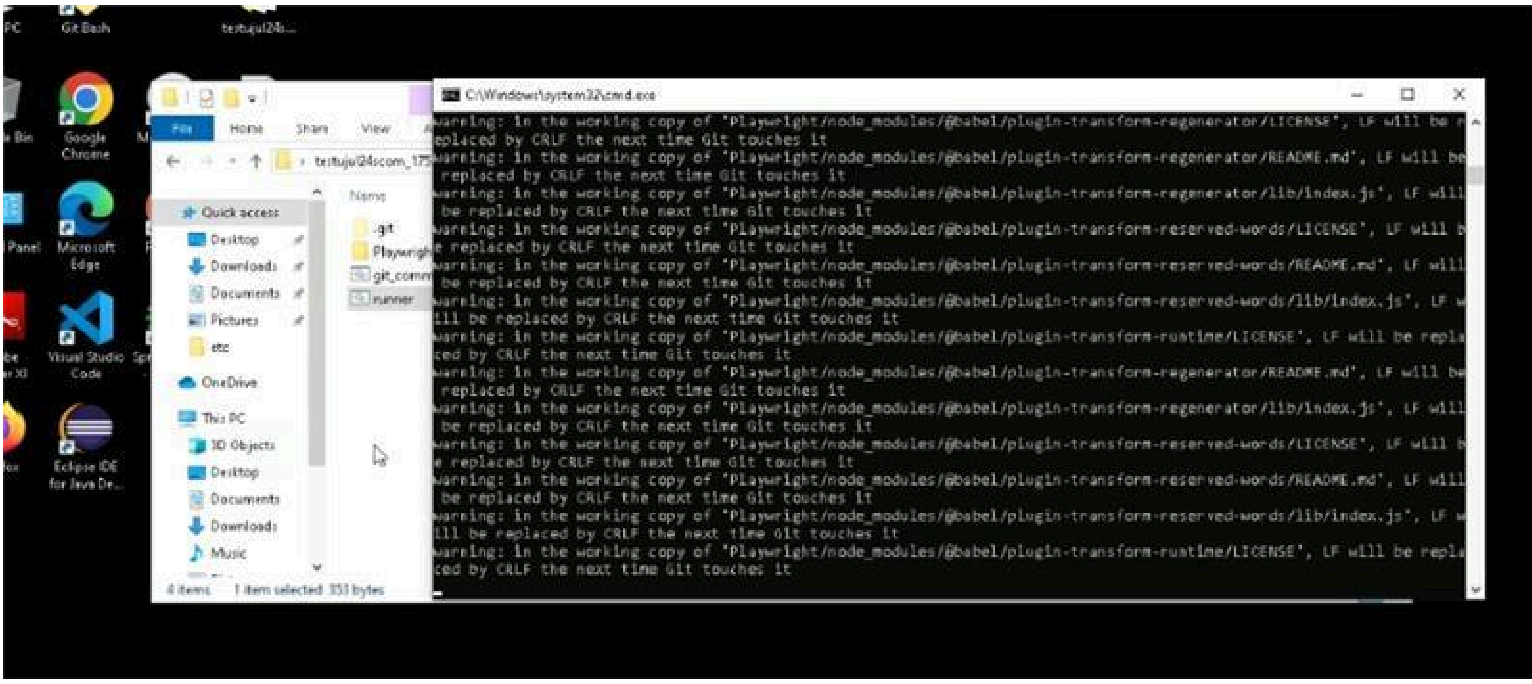
2. **Go into the Playwright folder and execute this “runner” file. This will keep pushing the code at**



Name	Date modified	Type	Size
.git	24-07-2025 18:45	File folder	
Playwright	24-07-2025 18:45	File folder	
git_commands	24-07-2025 18:45	Windows Batch File	1 KB
runner	24-07-2025 18:45	Windows Batch File	1 KB

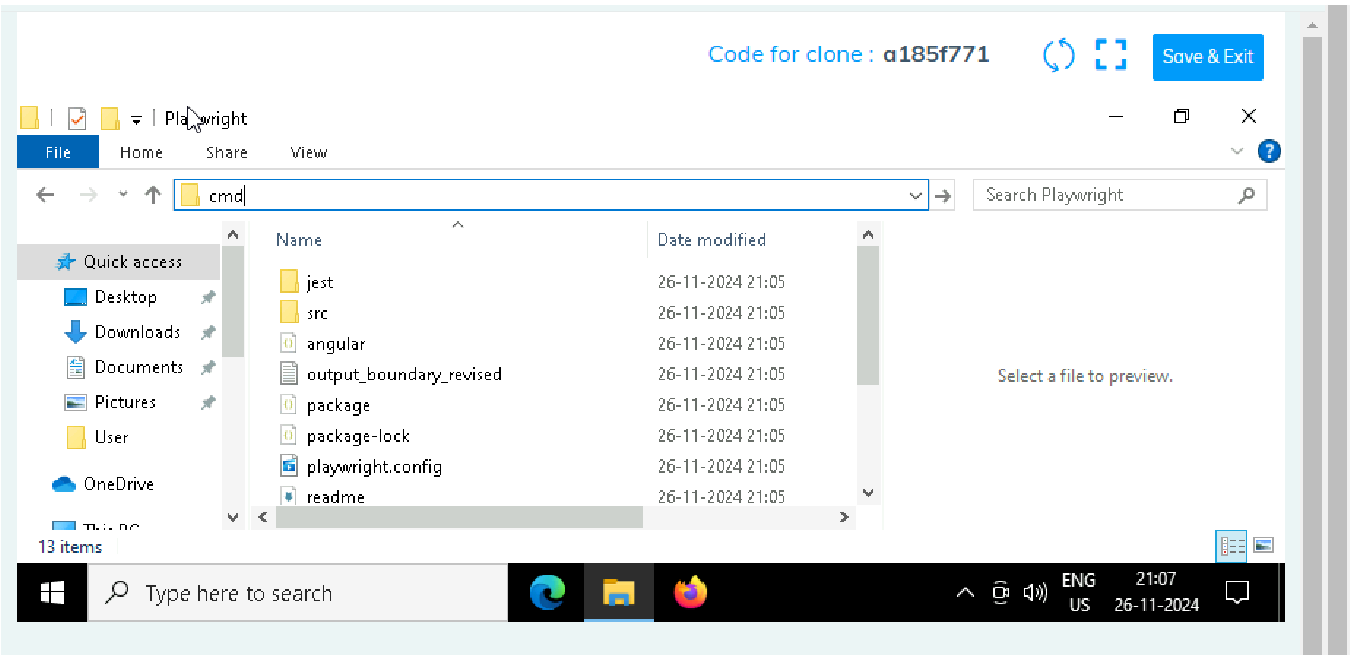
4 items   1 item selected 353 bytes

```
@echo off

REM Path to your batch
execute
set
"batch_file_path=git_co
"

REM Interval in seconds
seconds = 1 minute)
set interval=60

:loop
echo Executing batch fi
DATE% %TIME%
call "%batch_file_path%

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

goto loop
```



```
warning: in the working copy of 'Playwright/node_modules/@babel/plugin-transform-regenerator/LICENSE', LF will be r
eplaced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/@babel/plugin-transform-regenerator/README.md', LF will be
 replaced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/@babel/plugin-transform-regenerator/lib/index.js', LF will
 be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/@babel/plugin-transform-reserved-words/LICENSE', LF will b
e replaced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/@babel/plugin-transform-reserved-words/README.md', LF will
 be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/@babel/plugin-transform-reserved-words/lib/index.js', LF w
ill be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/@babel/plugin-transform-runtime/LICENSE', LF will be repla
ced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/@babel/plugin-transform-regenerator/README.md', LF will be
 replaced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/@babel/plugin-transform-regenerator/lib/index.js', LF will
 be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/@babel/plugin-transform-reserved-words/LICENSE', LF will b
e replaced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/@babel/plugin-transform-reserved-words/README.md', LF will
 be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/@babel/plugin-transform-reserved-words/lib/index.js', LF w
ill be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/@babel/plugin-transform-runtime/LICENSE', LF will be repla
ced by CRLF the next time Git touches it
```

3. **Go into the Playwright folder and Open command prompt with its location and use the command below: code .**

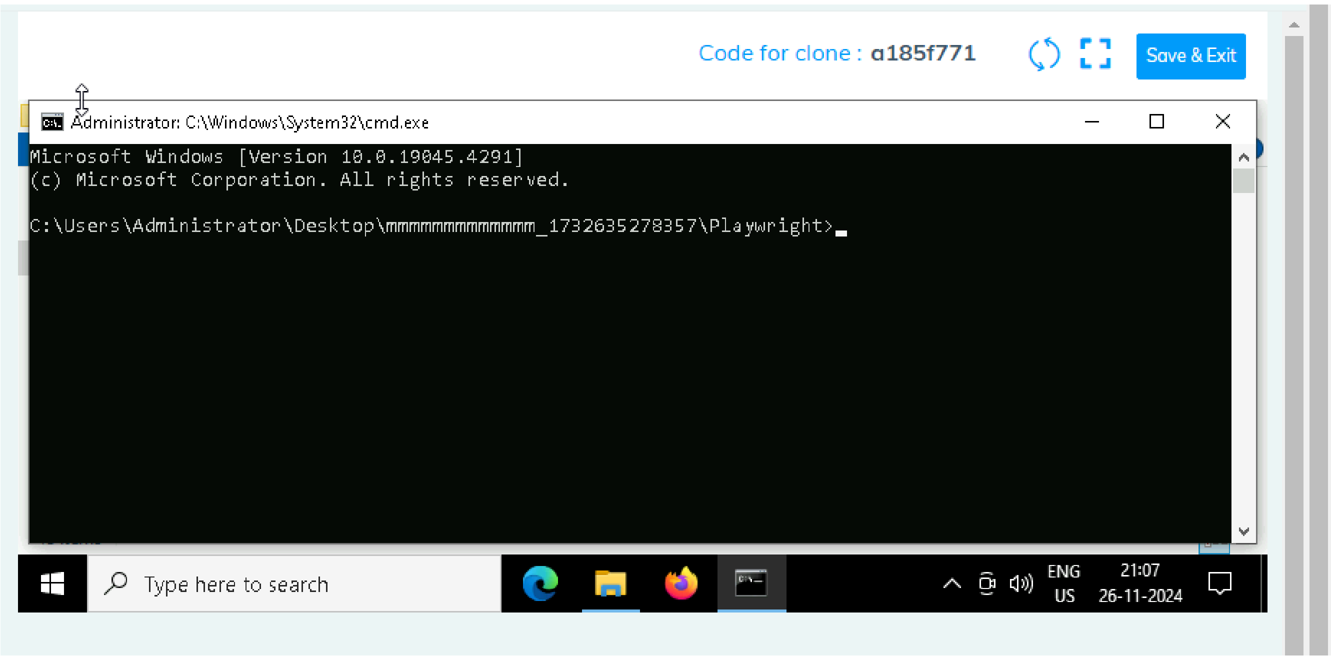


cmd

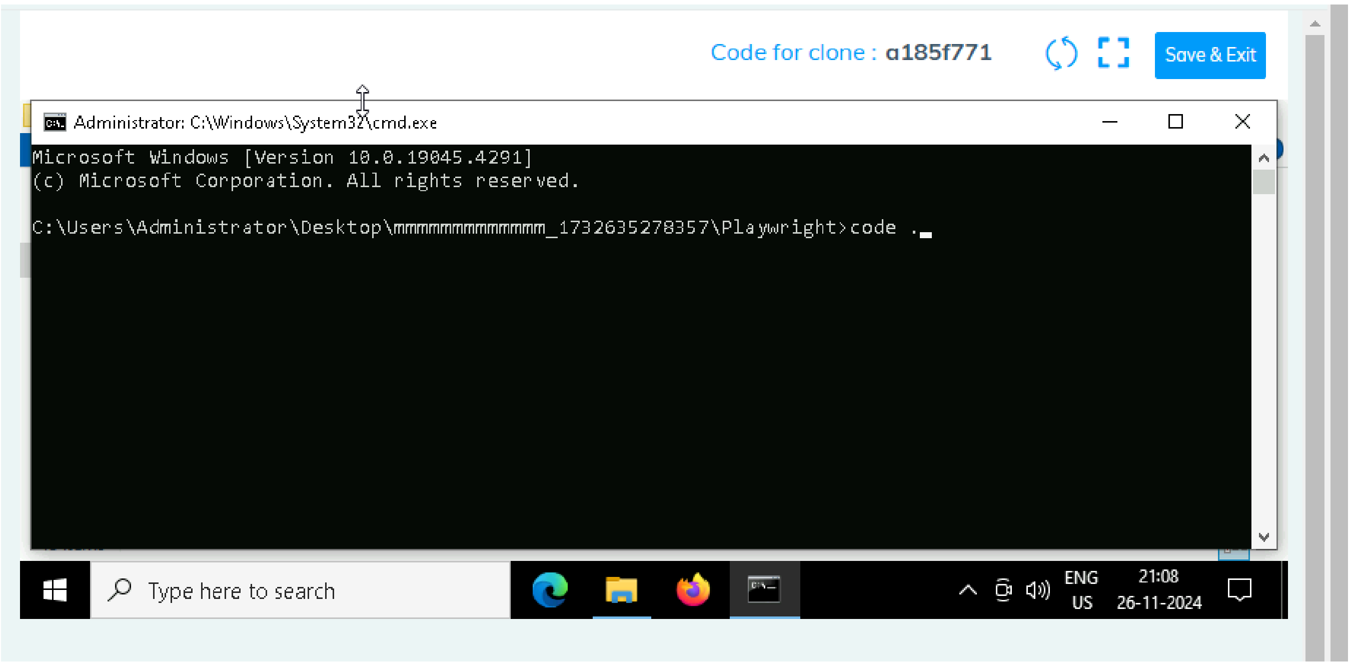
Name	Date modified
jest	26-11-2024 21:05
src	26-11-2024 21:05
angular	26-11-2024 21:05
output_boundary_revised	26-11-2024 21:05
package	26-11-2024 21:05
package-lock	26-11-2024 21:05
playwright.config	26-11-2024 21:05
readme	26-11-2024 21:05

Select a file to preview.

13 items

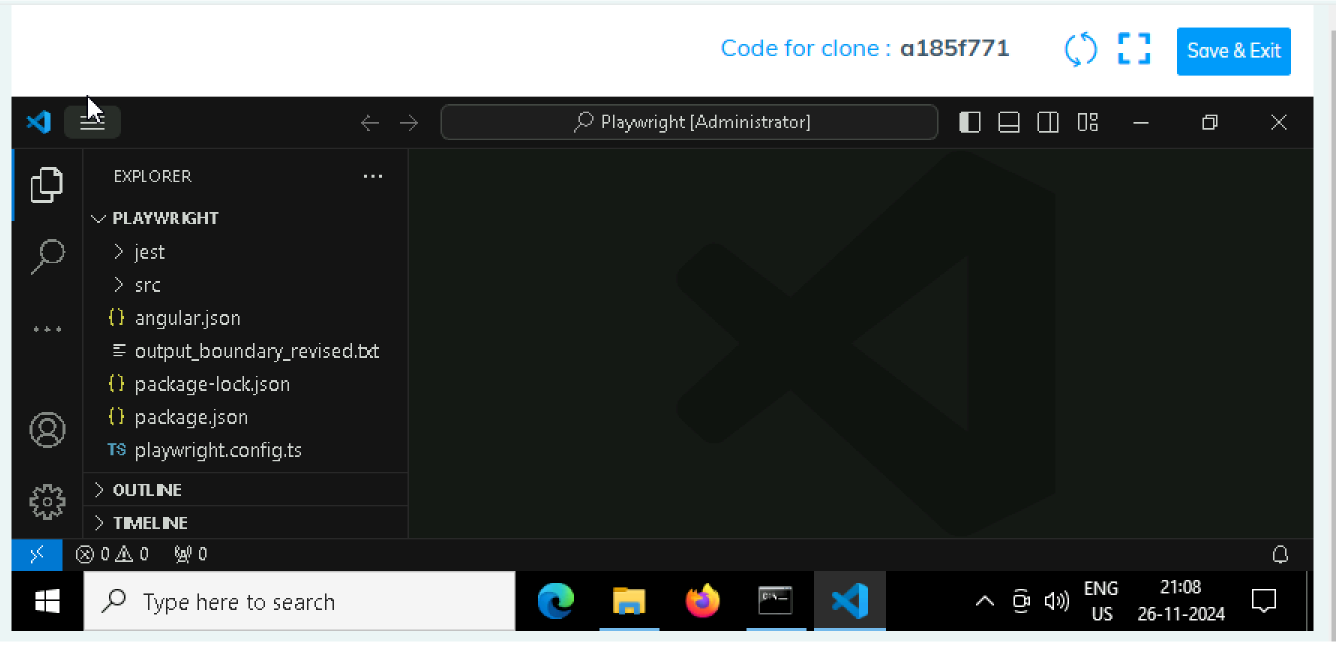

Code for clone : a185f771
Save & Exit
Administrator: C:\Windows\System32\cmd.exe
Type here to search
ENG
US
21:07
26-11-2024

```
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 .
```

4. **Once VsCode is open. Please open the terminal in Playwright folder:**



5. **Install all dependencies in the Playwright folder**

   path using: npm install

6. **Install playwright in the Playwright**

   folder path: npx playwright install

7. **Run the Tests in the Playwright folder path**:

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