YAKSHA HEALTH APP WITH TYPESCRIPT AND PLAYWRIGHT PL1-8

Usecase summary

Project Name:opensource-demo.orangehrmlive.com app – HR management system.

Use Case Summary: opensource-demo.orangehrmlive.com is designed to manage Employee record.

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 few Json files in Data folder like Login.json and TestData.Json

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

src\ pages	● DependentsPage ● 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 personal info attachment can be edited	1. Login using valid credentials 2. Navigate to "My Info". 3. Click edit on a personal info attachment. 4. Update the comment and save.	**Reference path** \src\pages\MyInfoPage **Methods:** editattachmentPersonaldetail()	The updated comment should be displayed in the attachment list
2	Verify membership attachment comment can be edited	1. Login using valid credentials 2. Navigate to "My Info" → Memberships. 3. Click the edit (pencil) icon for an attachment. 4. Enter a new comment and save.	**Reference path** \src\pages\MyInfoPage **Methods:** editMmbrAtchmntCmnt()	The updated comment should appear in the list.
3	Verify the name gets edited successfully	1. Login using valid credentials 2. Click on "My Info" tab 3. Clear the name present in the name inputbar 4. Enter the new name and hit save button 5. Hit refresh	**Reference path** \src\pages\MyInfoPage **Methods:** updateUniqueNAmeAndVerifyName() **CommonMethods file to highlight the element before performing any action on it. It takes locator as a parameter.**	Verify the name gets updated in the top right corner

4	Verify the 'My Info' tab's subtabs have unique URL	1. Login using valid credentials 2. Click on "My Info" tab 3. Click on top 3 items from the sidebar('personal details',contact details,emergency contact')	**Reference path** \src\pages\DependentsPage **Methods:** areMyInfoSubTabHrefsUnique()	Verify the url from the three is unique
5	Verify new Dependant could be added to the list	1. Login using valid credentials 2. Click on "My Info" tab 3. Navigate to 'Dependents' subtab 4. Fill name ,select 'other' from relationship 5. Enter the name of other relationship.	**Reference path** \src\pages\DependentsPage **Methods:** addDependent()	Verify the presence of the child in the list
6	Verify new Dependant 'Specify' inputbar only displays when 'Other' option is selected from the relationship dropdown	1. Login using valid credentials 2. Click on "My Info" tab 3. Navigate to 'Dependents' subtab 4. Fill name ,select 'other' from relationship 5. Enter the name of other relationship	**Reference path** \src\pages\DependentsPage **Methods:** selectOtherAndCheckSpecifyField()	Verify the specify inputbar is not present before entering the other in relationship inputbar
7	Verify the dependants could be edited from the list	1. Login using valid credentials 2. Click on "My Info" tab 3. Navigate to 'Dependents' subtab 4. Create a new dependant using previous steps 5. Click the edit icon and change the name of the dependant	**Reference path** \src\pages\DependentsPage **Methods:** editDependentNameFlow()	Verify the dependant details get updated

8	Verify the dependants could be deleted from the list	1. Login using valid credentials 2. Click on "My Info" tab 3. Navigate to 'Dependents' subtab 4. Create a new dependant using previous steps 5. Click the delete icon and verify the candidate gets deleted from the list.	**Reference path** \src\pages\DependentsPage **Methods:** deleteDependentFlow()	Verify the dependant details get deleted

9	Verify the sample pdf file could be uploaded in the attachments	1. Login using valid credentials 2. Click on "My Info" tab 3. Navigate to 'Dependents' subtab 4. Click on 'Add' Attachments button 5. Browse the pdf from the device 6. Click save button	**Reference path** \src\pages\DependentsPage **Methods:** UploadAttachmentInDependent()	Verify the pdf gets uploaded
10	Verify the Comment inputbar has limit on length	1. Login using valid credentials 2. Click on "My Info" tab 3. Navigate to 'Dependents' subtab 4. Click on 'Add' Attachments button 5. Click Comments inputbar and add really long text (eg. longer than 200 characters).	**Reference path** \src\pages\DependentsPage **Method:** CommentBarInputLimit()	Verify the warning message like: 'Should not exceed 200 characters' is displayed when the character limit is exceeded.

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 behaviour meets expected results.
- **End-to-End Testing**: Automating real user interactions and validating overall app functionality.

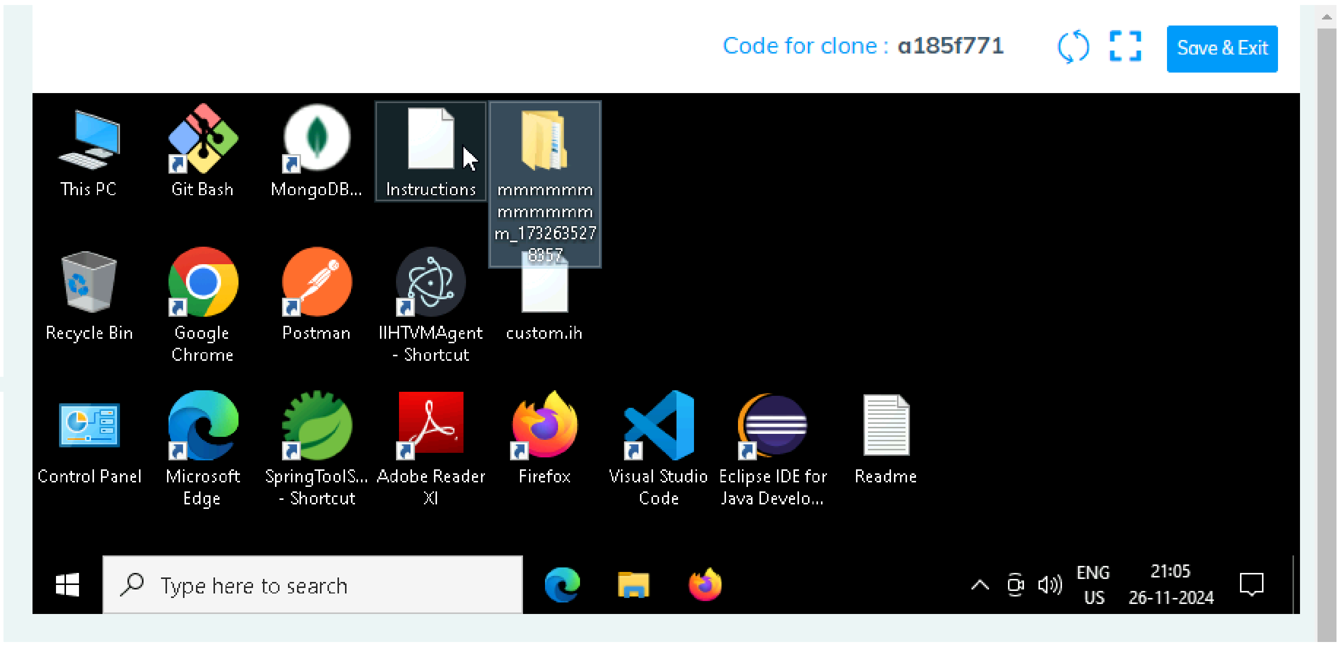
IMPLEMENTATION/FUNCTIONAL REQUIREMENT

1.1 Code Quality/Optimizations

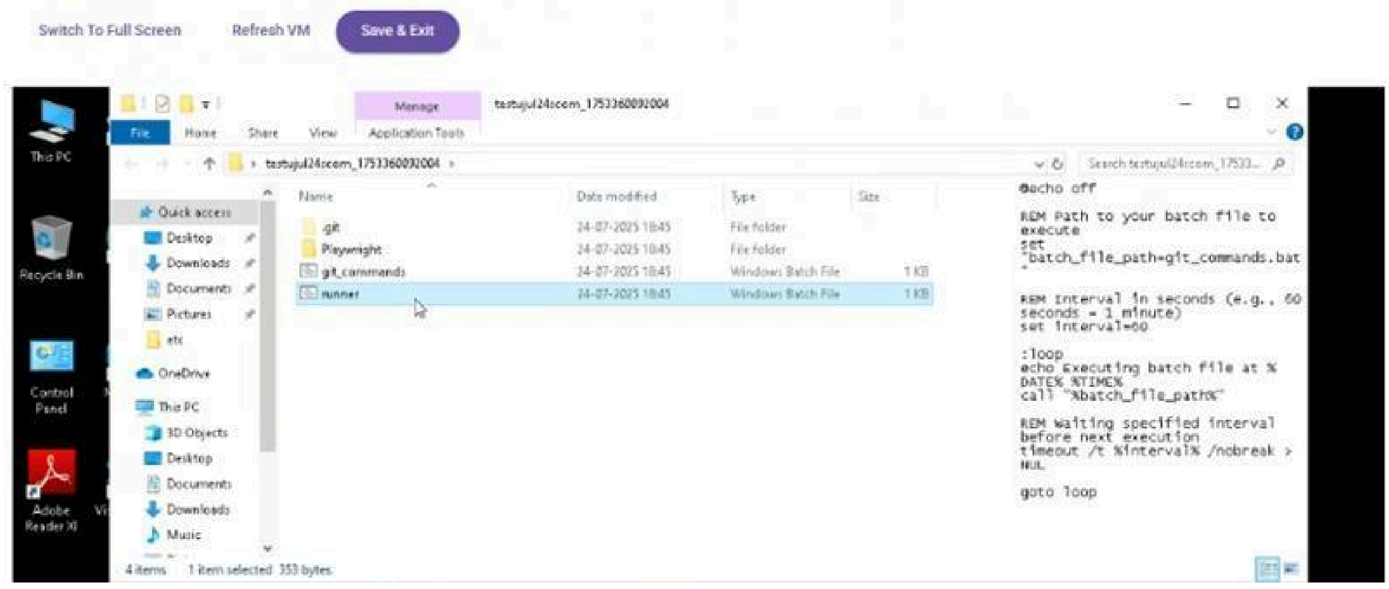
1. Associates should have written clean code that is readable.
2. Associates need to follow SOLID programming principles.

Execution Steps:
Steps for Execution:

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



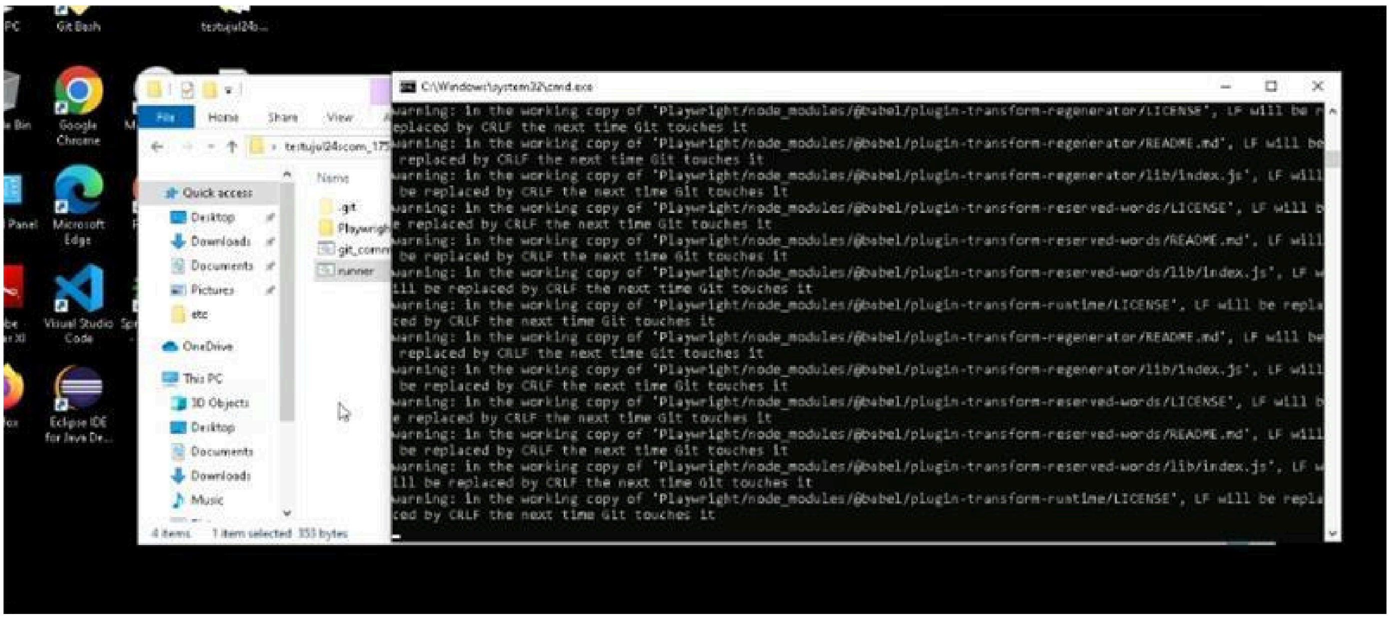
2. Execute this “runner” file. This will keep pushing the code at regular intervals



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

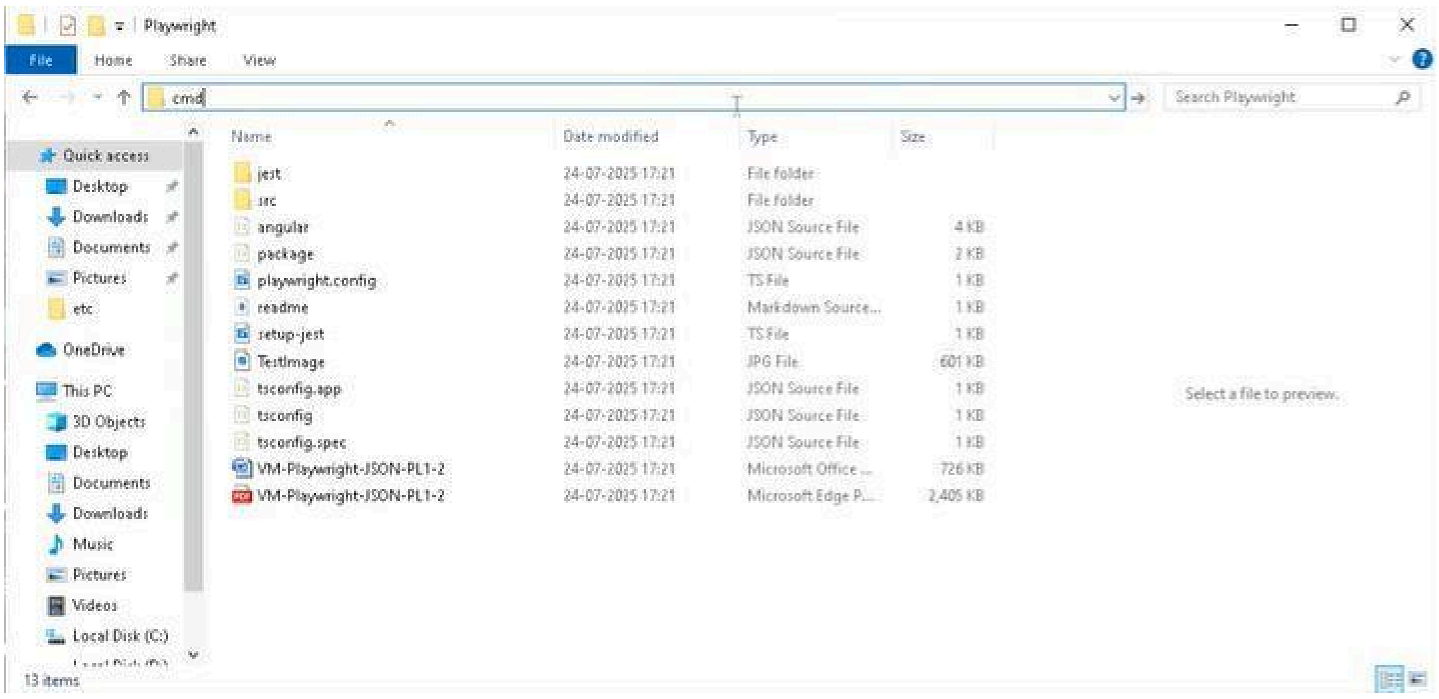
4 items   1 item selected 353 bytes

```
@echo off

REM Path to your batch 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
```

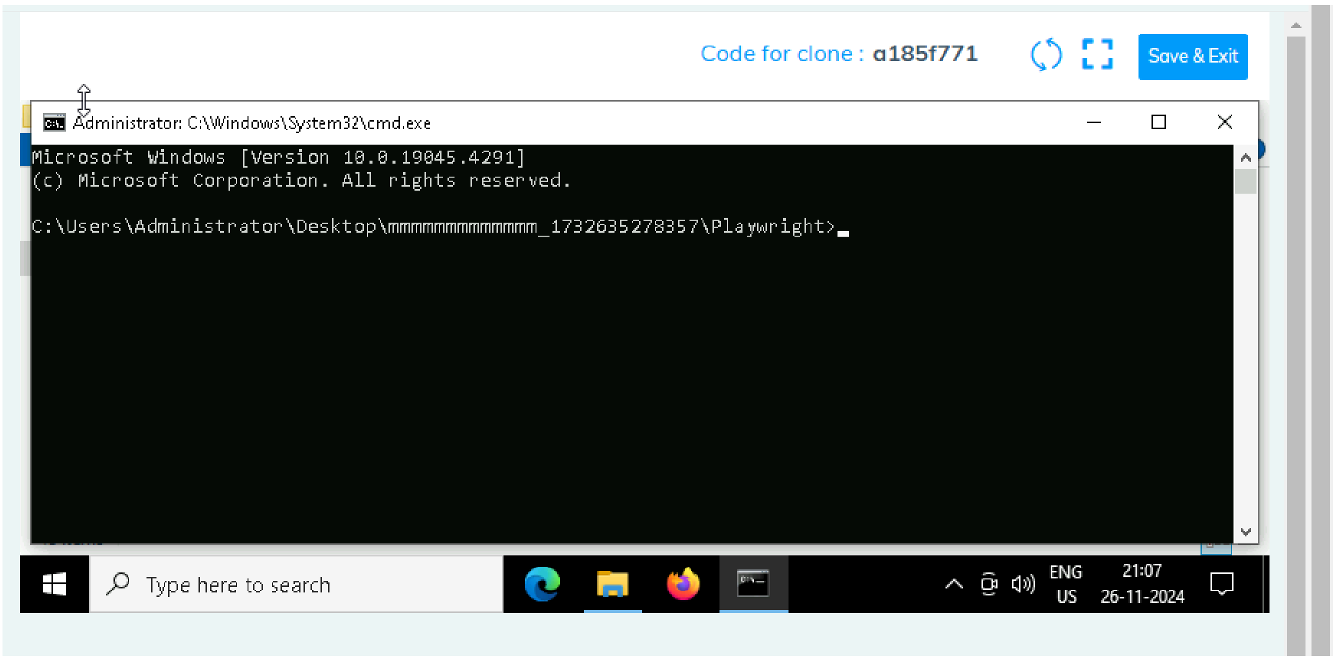


```
C:\Windows\system32\cmd.exe
warning: in the working copy of 'Playwright/node_modules/@babel/plugin-transform-regenerator/LICENSE', LF will be r
eplaced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/@babel/plugin-transform-regenerator/README.md', LF 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 below command: code .

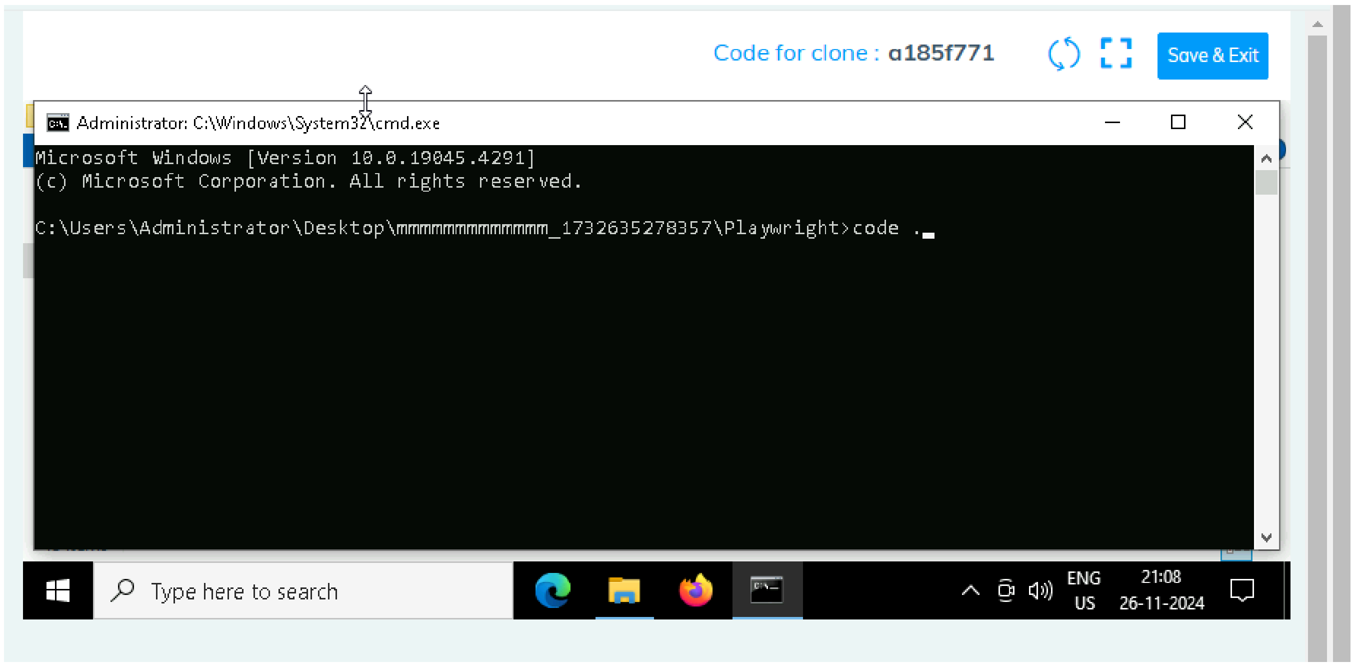


Name	Date modified	Type	Size
jest	24-07-2025 17:21	File folder	
src	24-07-2025 17:21	File folder	
angular	24-07-2025 17:21	JSON Source File	4 KB
package	24-07-2025 17:21	JSON Source File	2 KB
playwright.config	24-07-2025 17:21	TS File	1 KB
readme	24-07-2025 17:21	Markdown Source...	1 KB
setup-jest	24-07-2025 17:21	TS File	1 KB
TestImage	24-07-2025 17:21	JPG File	601 KB
tsconfig.app	24-07-2025 17:21	JSON Source File	1 KB
tsconfig	24-07-2025 17:21	JSON Source File	1 KB
tsconfig.spec	24-07-2025 17:21	JSON Source File	1 KB
VM-Playwright-JSON-PL1-2	24-07-2025 17:21	Microsoft Office ...	726 KB
VM-Playwright-JSON-PL1-2	24-07-2025 17:21	Microsoft Edge P...	2,405 KB



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

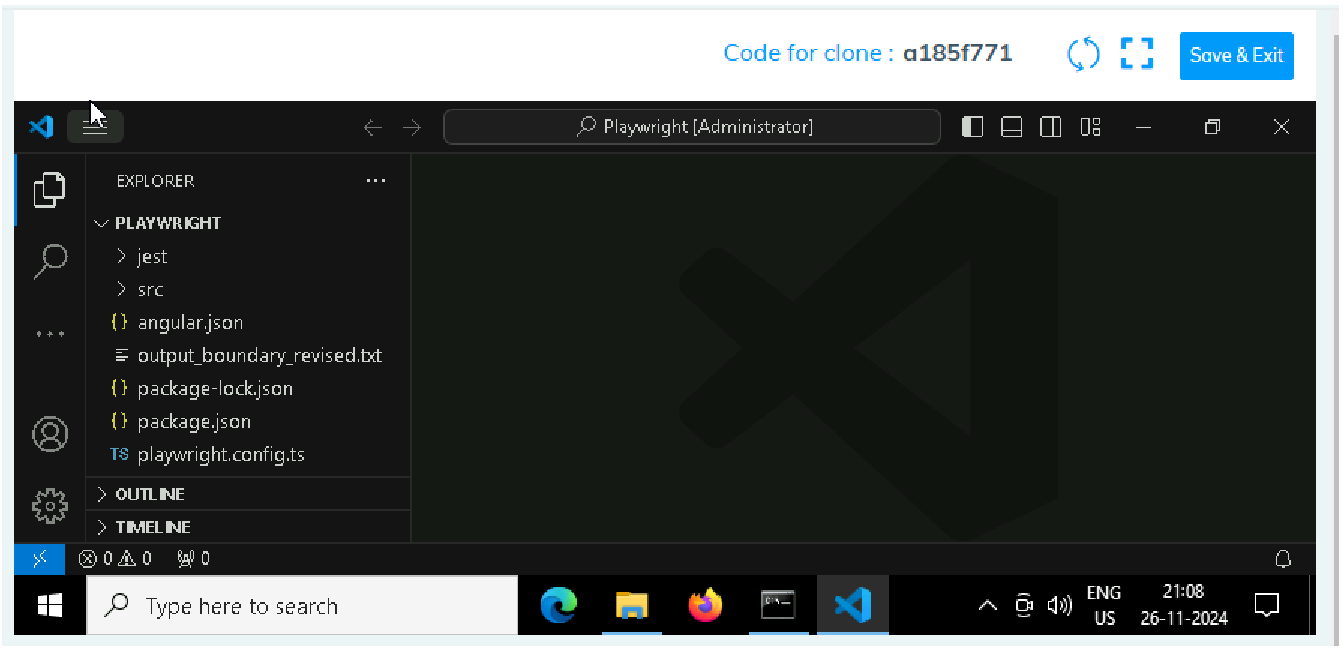
```
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

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
```