YAKSHA HEALTH APP WITH TYPESCRIPT AND PLAYWRIGHT PL1-7

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:/opensource-demo.orangehrmlive.com

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.json	1. Contains data to read from json file.

src\ pages	• 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 New 'Qualification' could be added to the record of user	Login using valid credentials 1. Click on "My Info" tab 2. Click 'Qualification' from the sidebar 3. Click on 'Add' work experience button	**Reference path** \src\pages\MyinfoPage **Method** addQualification() **You can use highlightElement method present in CommonMethods file to highlight the element before performing any action on it. It takes locator as a parameter.**	Verify the new qualification gets added to the list of records
2	Verify New 'Qualification' added could be edited from the record of use	1. Login using valid credentials 2. Click on "My Info" tab 3. Click 'Qualification' from the sidebar 4. Click on 'edit icon' button 5. Fill the new record to the company and job title	**Reference path** \src\ pages**MyinfoPage** **Method** editQualification() **You can use highlightElement method present in CommonMethods file to highlight the element before performing any action on it. It takes locator as a parameter.**	Verify the updated New record from the list of records
3	Verify the qualification could be deleted	1. Login using valid credentials 2. Click on "My Info" tab 3. Click 'Qualification' from the sidebar 4. Click on 'delete icon' button 5. Confirm the Delete	**Reference path** \src\ pages**MyInfoPage** **Method** deleteQualification()	Verify the qualification gets deleted from the record

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

4	Verify the qualification select delete functionality	1. Login using valid credentials 2. Click on "My Info" tab 3. Click 'Qualification' from the sidebar 4. Select top 3 from the list 5. Click select deleted and confirm delete	**Reference path** \src\ pages**MyInfoPage** **Method** deleteMultiple() **You can use highlightElement method present in CommonMethods file to highlight the element before performing any action on it. It takes locator as a parameter.**	Verify the selected qualification gets deleted successfully
5	Verify New 'Educati on' could be added to the record of user.	1. Login using valid credentials 2. Click on "My Info" tab 3. Click 'Qualification' from the sidebar 4. Click the 'Add' button 5. Fill the form and click save button	**Reference path** \src\ pages**MyinfoPage** **Method** addEducation() **You can use highlightElement method present in CommonMethods file to highlight the element before performing any action on it. It takes locator as a parameter.**	Verify the newly added education gets displayed in the records
6	Verify New 'Education' added could be edited from the record of user	1. Login using valid credentials 2. Click on "My Info" tab 3. Click 'Qualification' from the sidebar 4. Click the 'Edit' icon button for desired entry 5. Fill the form(year) and hit save	**Reference path** \src\pages**MyinfoPage** **Method** editEducation()	Verify the year field gets replaced in the records
7	Verify New 'Language' could be added to the record of user	1. Generate a unique comment. 2. Navigate to Language section. 3. Select options from dropdowns. 4. Enter comment and save. 5. Assert comment appears in table.	**Reference path** \PageObjects\Pages**MyinfoPage** **Method** addLanguage()	Verify that the newly added comment is present in the language table

8	Verify the language should be deleted to the record of user	1. Add a new language entry with a unique comment. 2. Delete the entry using the comment as reference. 3. Retrieve updated language comments. 4. Verify the comment is no longer present.	**Reference path** \src\pages**MyInfoPage** **Method** deleteLanguage() **You can use highlightElement method present in CommonMethods file to highlight the element before performing any action on it. It takes locator as a parameter.**	Language should be deleted.
9	Verify the 'Skills' could be added to the record	1. Login using valid credentials 2. Click on "My Info" tab 3. Click 'Qualification' from the sidebar 4. Click on 'add' skill button 5. Fill the details and hit save button	**Reference path** \src\ pages**MyinfoPage** **Method** addSkills() **You can use highlightElement method present in CommonMethods file to highlight the element before performing any action on it. It takes locator as a parameter.**	Verify 'Skills' gets added to the list of records
10	Verify the 'Skills' could be edited from the record	1. Login using valid credentials 2. Click on "My Info" tab 3. Click 'Qualification' from the sidebar 4. Click on 'edit' icon on desired skill 5. Fill the Form and hit save button **Note :- If you cannot find the locator for "Proceed" button, please try to keep the page at 75%-80%.**	**Reference path** \src\pages**MyInfoPage** **Method** editSkills() **You can use highlightElement method present in CommonMethods file to highlight the element before performing any action on it. It takes locator as a parameter.**	Verify 'skill' gets edited successfully from the list of records

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:/opensource-demo.orangehrmlive.com

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.

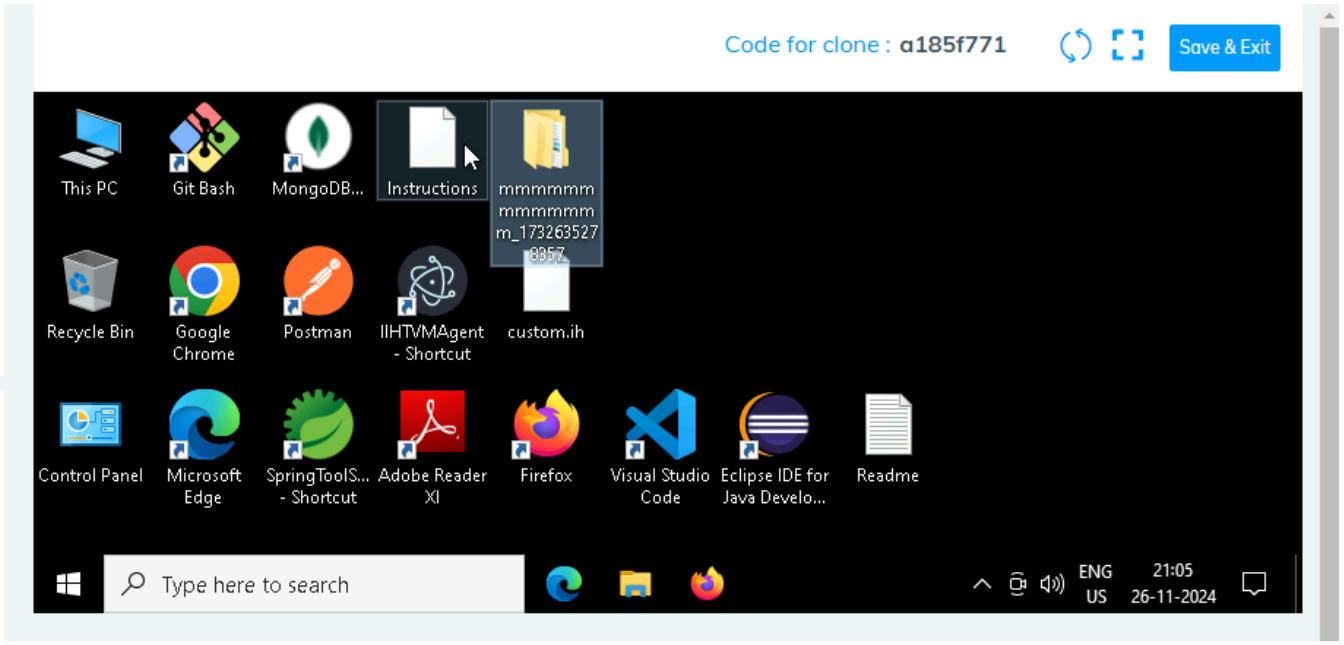
IMPLEMENTATION/FUNCTIONAL REQUIREMENT

1.1 Code Quality/Optimizations

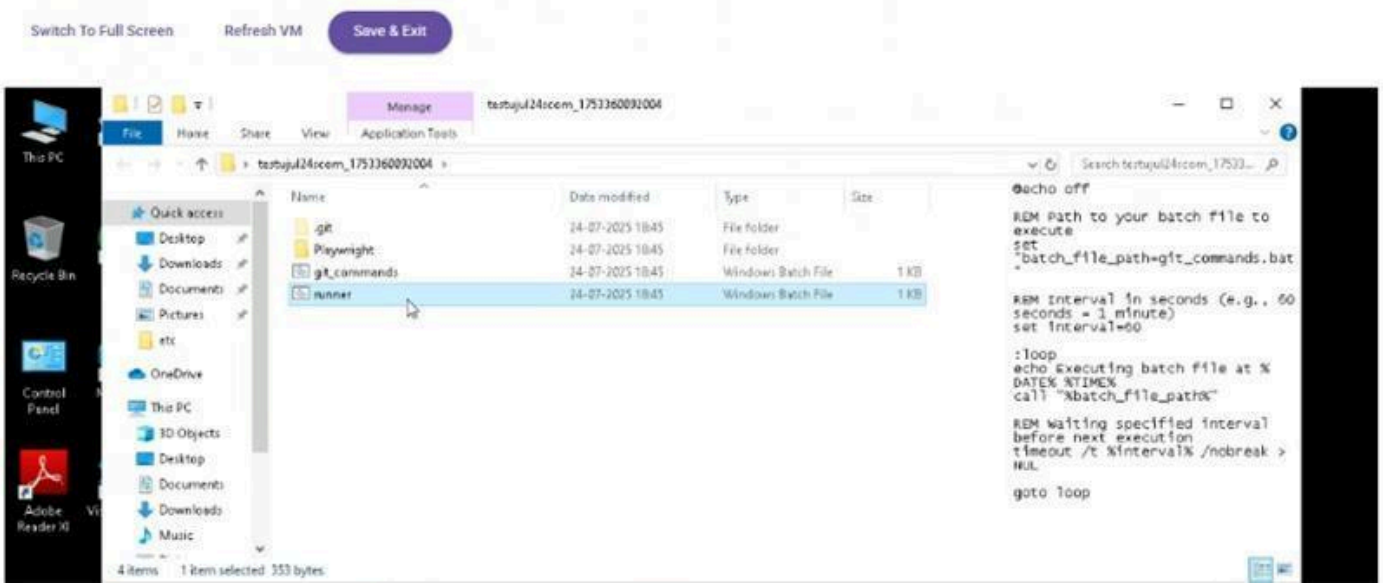
1. Associates should have written clean code that is readable.
2. Associates need to follow SOLID programming principles.

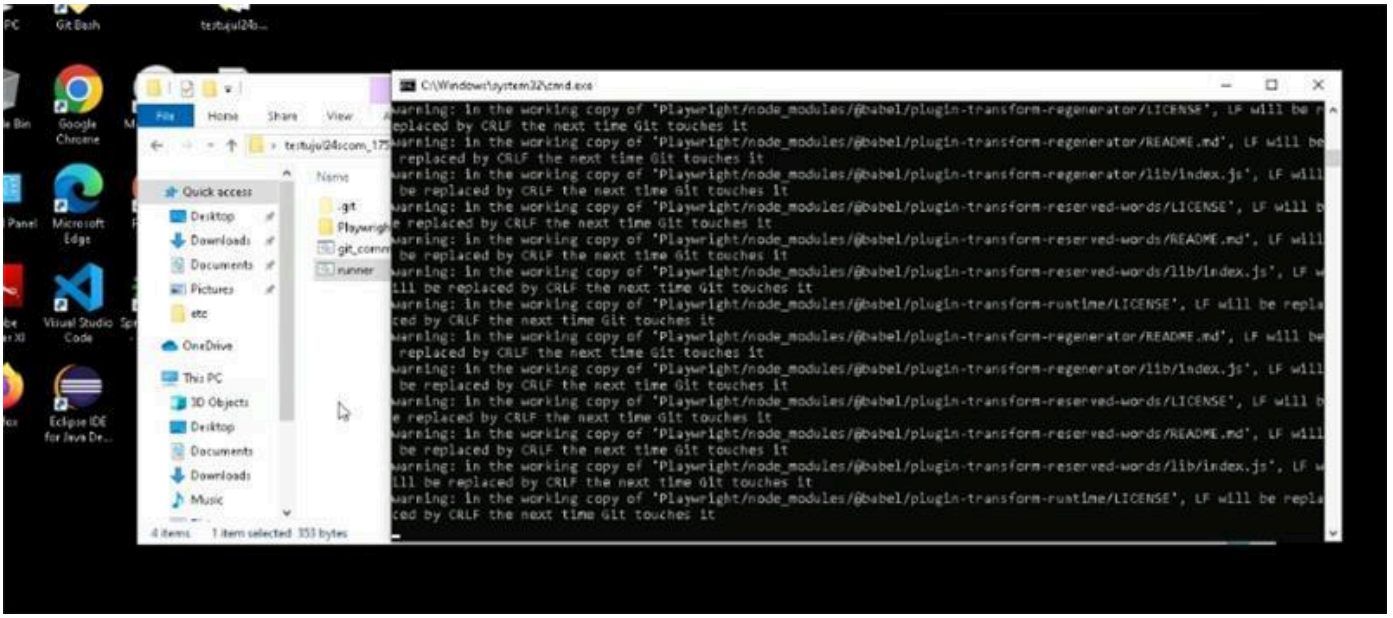
Execution Steps:
Steps for Execution:

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

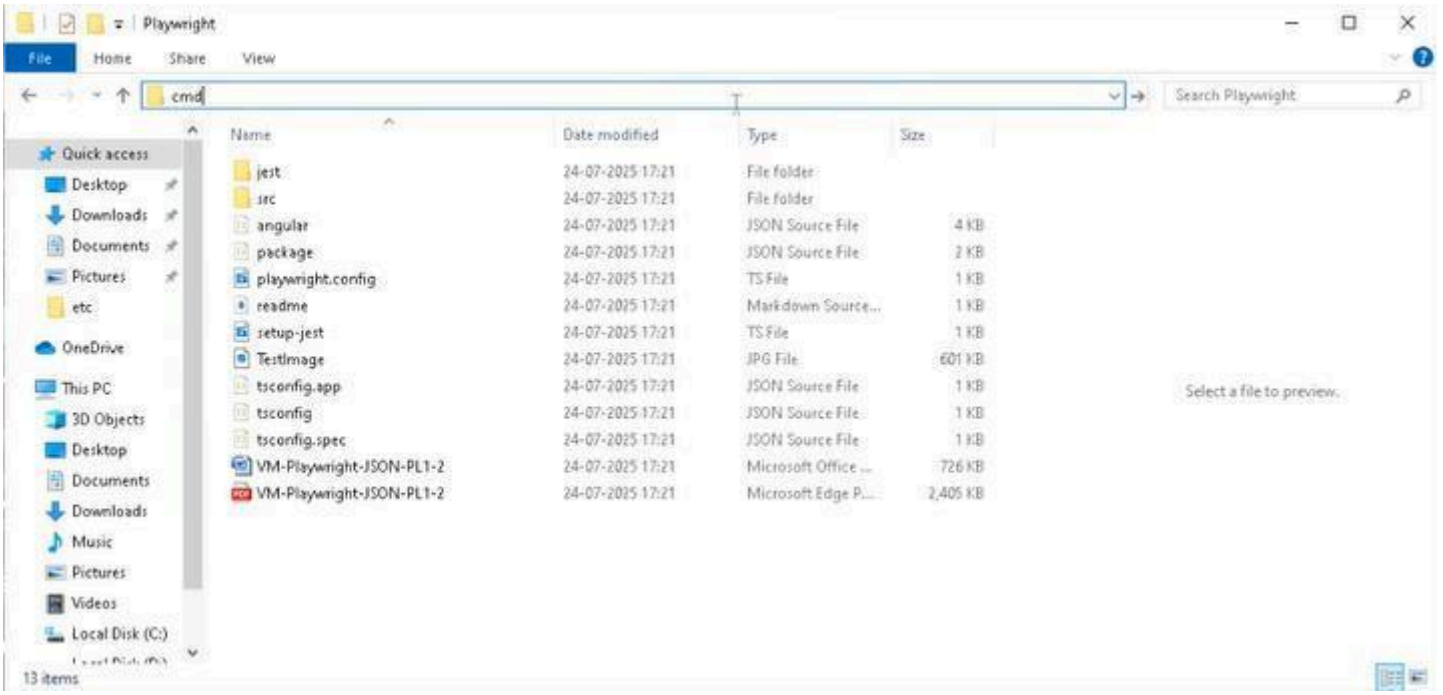


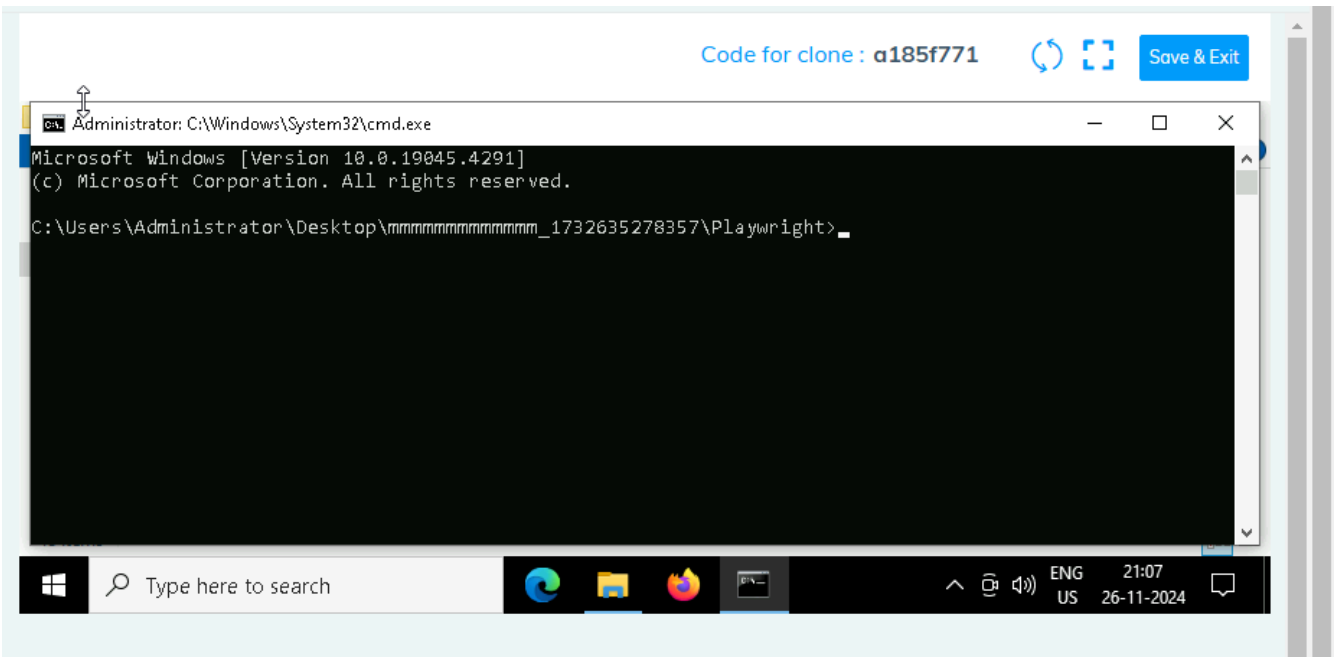
2. Execute this “runner” file. This will keep pushing the code at regular intervals

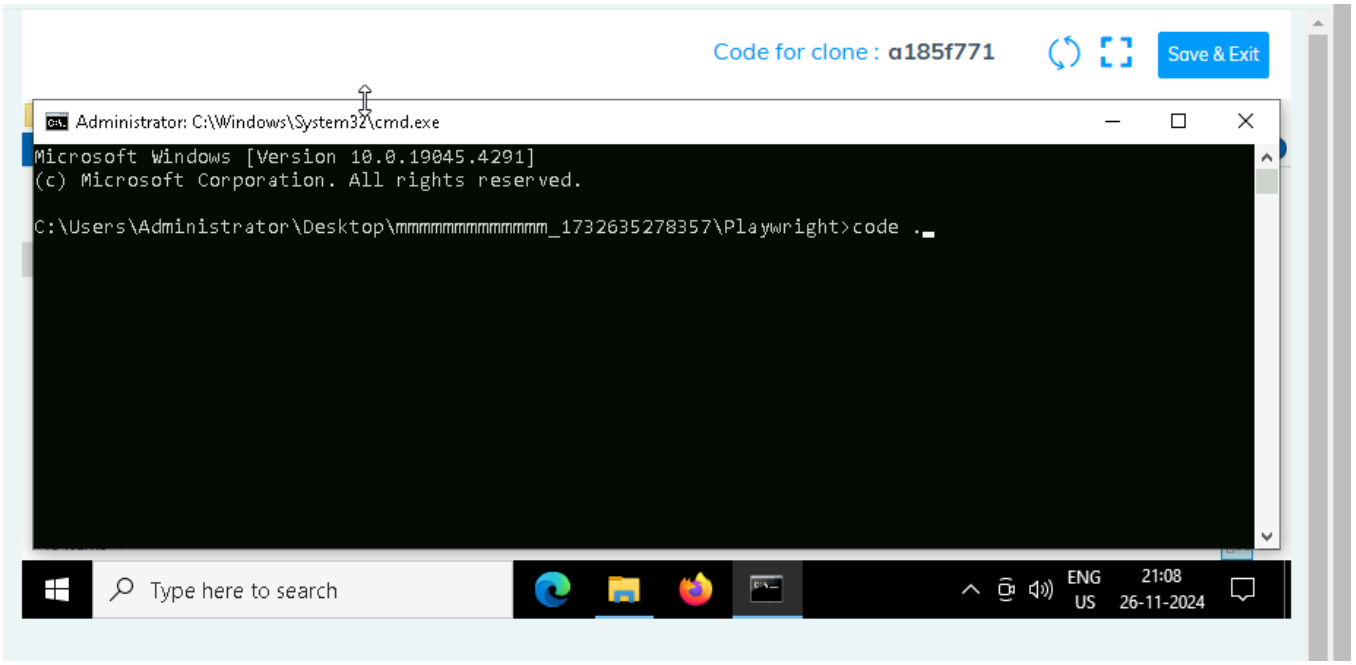




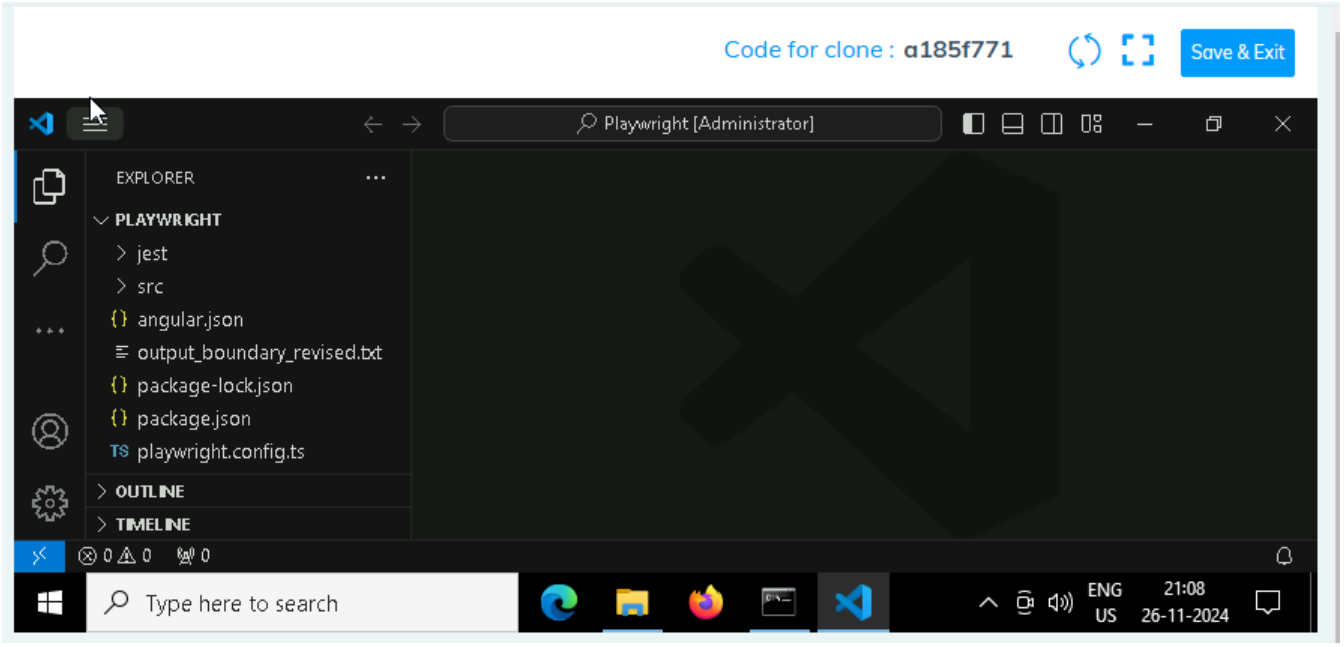
3. Go into the Playwright folder and Open command prompt with its location and use below command: code .







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