YAKSHA HEALTH APP WITH TYPESCRIPT AND PLAYWRIGHT

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

|  |  |  |
| --- | --- | --- |
| src\ pages | * LoginPage * ClaimPage * 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 | 1. Login using valid credentials  2. Click on "My Info" tab  3. Click 'Qualification' from the sidebar  4. Click on 'Add' workexperience button  **It must implement this login**  **functionality at first and then implement any other test case.** | **Reference path**  \src\pages\MyinfoPage  **methods**  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** |
|  |  |  | **Reference path** |  |
| 2 | Verify New 'Qualification' 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 on 'edit icon' button**  **5. Fill the new record to the company and job title** | \src\ pages\**MyinfoPage**  **methods**  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 New record from the list of records** |
|  |  |  | **Reference path** |  |
| 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 | \src\ pages\**MyinfoPage**  **Methods:- deleteQualification()** | Verify the qualification gets deleted from the record |
|  |  |  | **You can use highlightElement method present in** |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case No.** | **Test Case Name** | **Test Steps to be performed** | | **Path & Method Used** | **Expected Result** |
|  |  |  | | **CommonMethods file to highlight the element before performing any action on it. It takes locator as a parameter.** |  |
|  |  | 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** |  |
| 4 | Verify the qualification select delete functionality | **Methods: -**  **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 |
|  |  |  | | **Reference path** |  |
| 5 | **Verify New 'Education' 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 | | \src\ pages\ **myinfoPage**  **methods**  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 Event tab URL in Claim configuration** | 1. Login with valid credentials. 2. Click the Claim tab. 3. Open Configuration sub-tab. 4. Click the Event sub-tab | | Reference path  \src\pages\**ClaimPage**  methods: **clickClaimButton**() | **Page URL contains eventsPage URL contains ‘events’** |
| 7 | Verify Event creation in Claim configuration | **1. Login with valid credentials. 2. Click the Claim tab. 3. Open Configuration sub-tab. 4. Click the Event sub-tab. 5. Click the Add button. 6. Fill event details and submit..** | | **Reference path**  \**PageObjects\Pages\ClaimPage** **methods**: addevents() | **A newly added event appears in the event list.** |
| 8 | **Verify Event Deletion in Claim Configuration** | 1. Login with valid credentials. 2. Click the **Claim** tab. 3. Open **Configuration** sub-tab. 4. Click the **Event** sub-tab. 5. Locate the target event in the list. 6. Click the **Delete** icon/button for that event. | | **Reference path**    \src\pages\**ClaimPage**  **Methods: deleteEvents**() | Verify the event is no longer present in the list. |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case No.** | **Test Case Name** | **Test Steps to be performed** | **Path & Method Used** | **Expected Result** |
|  |  | **1.Login.**  **2. Go to Claim → Configuration → Event. 3. Find a target event. 4. Click Edit toggle. 5. Turn off Active. 6. Save button** | **Reference path** |  |
|  |  | \src\ pages\**ClaimPage** |  |
|  |  | **methods** |  |
| 9 | Event Inactive in Claim Configuration | inactivateEvents()  **You can use highlightElement method present in CommonMethods file to highlight the element before performing any action on it. It takes locator as** | Event status changes to **Inactive**. |
|  |  | 1. Login with valid credentials.   2. Navigate to the Claim tab.   3. Open Configuration sub-tab.  4. Select Expense Types.  5. Add a new expense type with a unique name | **a parameter.**  **Reference path**  \src\pages\**ClaimPage** |  |
| 10 | Verify that a new Expense Type can be added |  | **methods**  addExpense () | The added expense type should be visible in the expense type list. |
|  |  | **Note :- If you cannot find the locator for “Proceed” button, please try to keep the page at 75%-80%.** | **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.** |  |
| 11 | Verify the 'Skills' could be deleted from the record | 1. Login using valid credentials  2. Click on "My Info" tab  3. Click 'Qualification' from the sidebar  4. Hit delete button and confirm**Note: - If you cannot find the locator for “Proceed” button, please try to keep the page at 75%-80%.** | Reference path  \src\pages\MyinfoPage  Methods: deleteskills()  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 deleted succesfully from the list of records |
| 12 | Verifying the 'Skills' could be selected 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 of record and hit delete selected  5. Confirm delete  **Note: - If you cannot find the locator for “Proceed” button, please try to keep the page at 75%-80%.** | **Reference path**  \**src\pages\myinfoPage**  Methods: **selectDeleteSkills**()  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 'skills' gets deleted successfully from the list of records. |
| 13 | Verify that an Expense Type can be edited | 1. Login with valid credentials.   \* 2. Navigate to Claim tab.   \* 3. Open Configuration sub-tab.   \* 4. Select Expense Types.   \* 5. Edit an existing expense type by toggling the edit option.   \* 6. Update the details and save. **Note**: **- If you cannot find the locator for “Proceed” button, please try to keep the page at 75%-80%.** | **Reference path**  \**src\pages\ClaimPage**  Methods: **editExpense**()  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. | The edited expense type should display the updated details in the expense type list |
| 14 | Verify that an Expense Type can be deleted | 1. Login.  2. Navigate to the Claim tab.  3. Open Configuration sub-tab.   4. Select ExpenseTypes.   5. Click the delete (trash) icon for a specific expense type.   6. Confirm the deletion.  **Note: - If you cannot find the locator for “Proceed” button, please try to keep the page at 75%-80%.** | **Reference path**  \**src\pages\ClaimPage**  **Methods: deleteExpense()**  You can use highlight Element method present in Common Methods file to highlight the element before performing any action on it. It takes locator as a parameter. | The selected expense type should not appear in the expense type list after deletion. |
| 15 | Verify that multiple Expense Types can be selected and deleted | 1. Login with valid credentials.  2. Navigate to Claim tab.  3. Open Configuration sub-tab.  4. Select Expense Types.  5. Add expense types for deletion.  6. Select checkboxes for expense types.  7. Click "Delete Selected".  8. Confirm the deletion.   **Note: - If you cannot find the locator for “Proceed” button, please try to keep the page at 75%-80%.** | **Reference path**  \**src\pages\ClaimPage**  **Methods: -multipleExpenseDel()**  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. | The selected expense types should be removed from the expense type list |

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. **Code Quality/Optimizations**

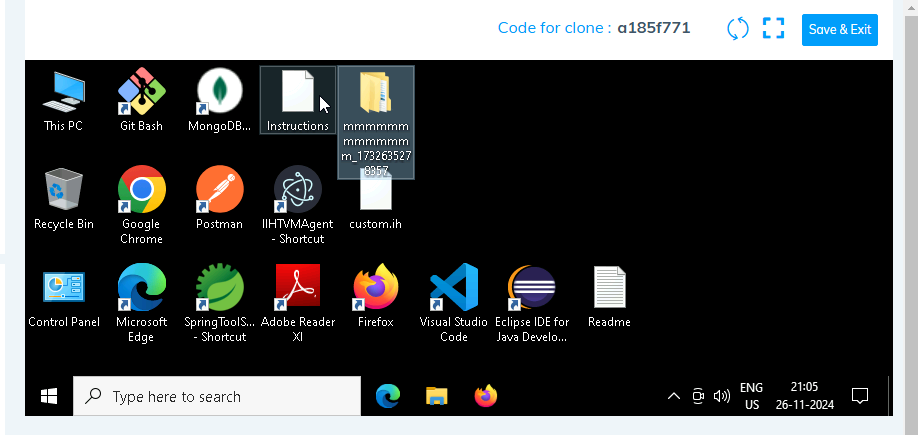
# Associates should have written clean code that is readable.

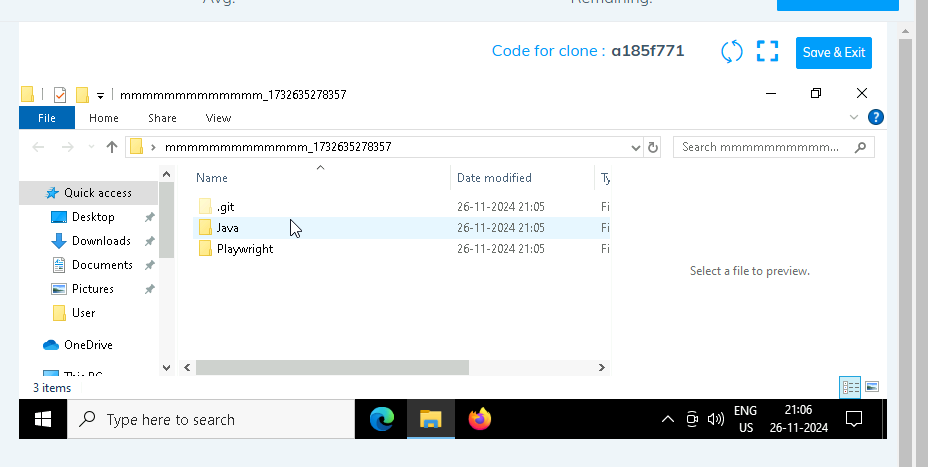
* + 1. Associates need to follow SOLID programming principles.

**Execution Steps:**

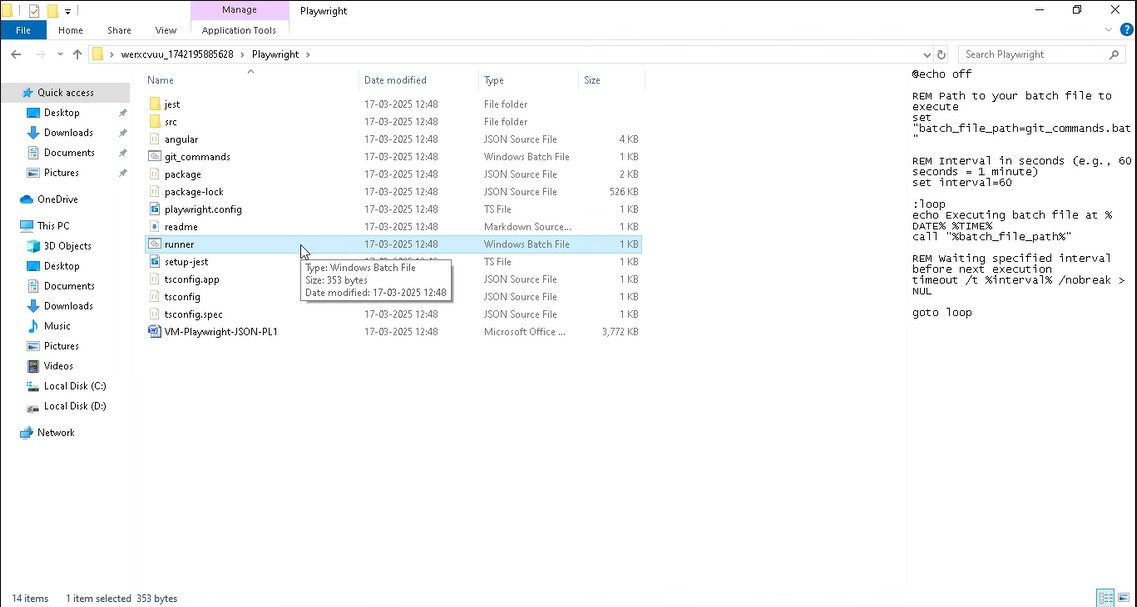
**Steps for Execution:**

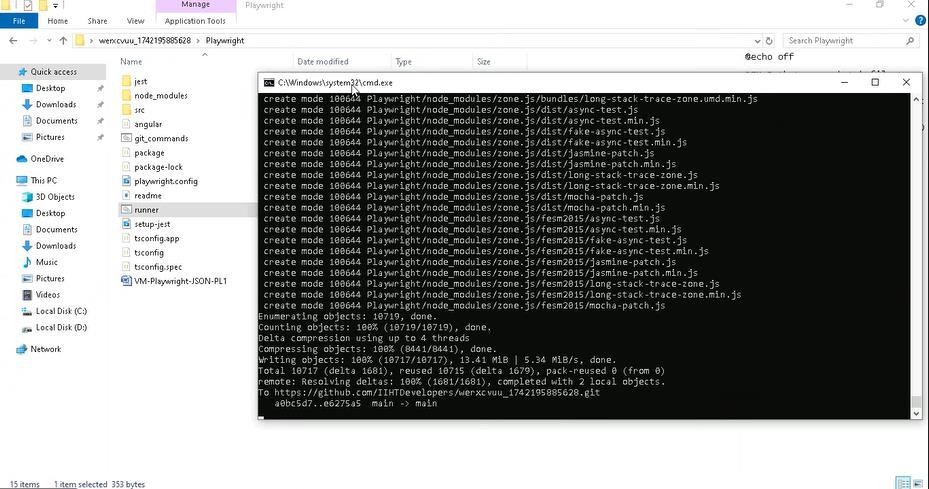
1. **Please open the folder created on desktop with the email name you used to login.**

****

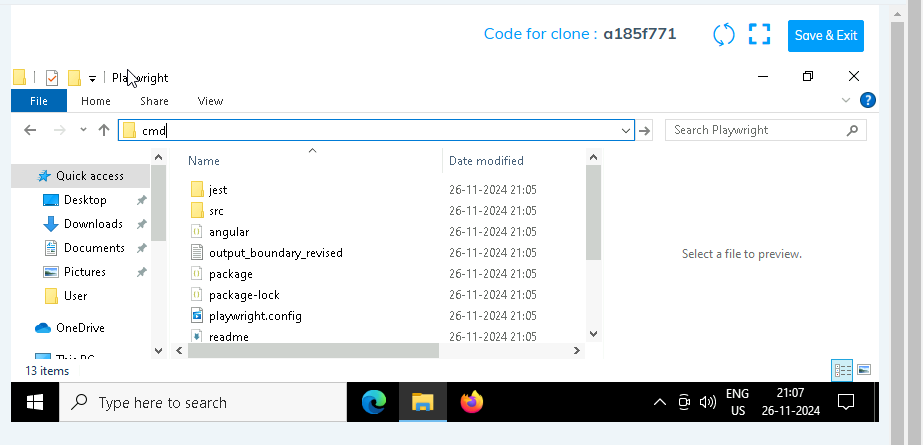


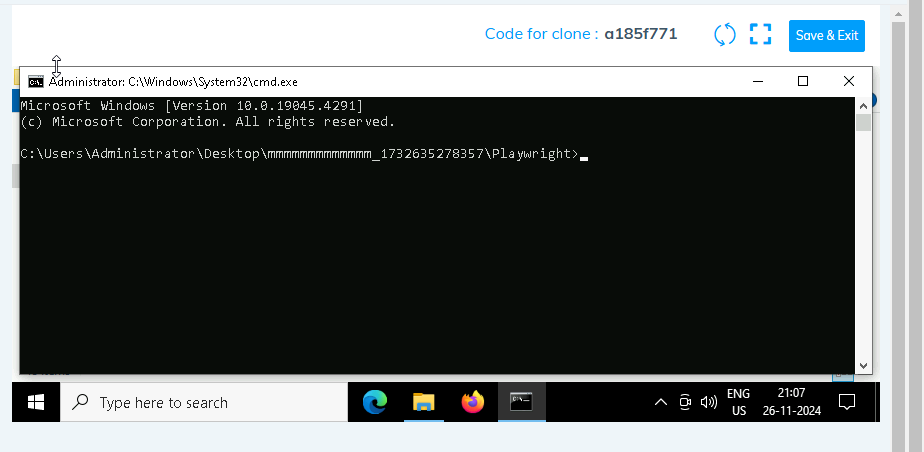
1. **Go into the Playwright folder and execute this “runner” file. This will keep pushing the code at regular intervals**

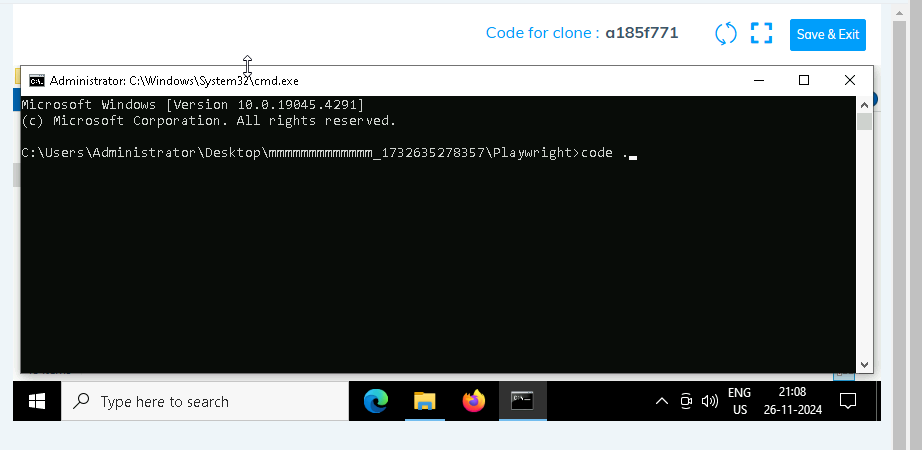


****

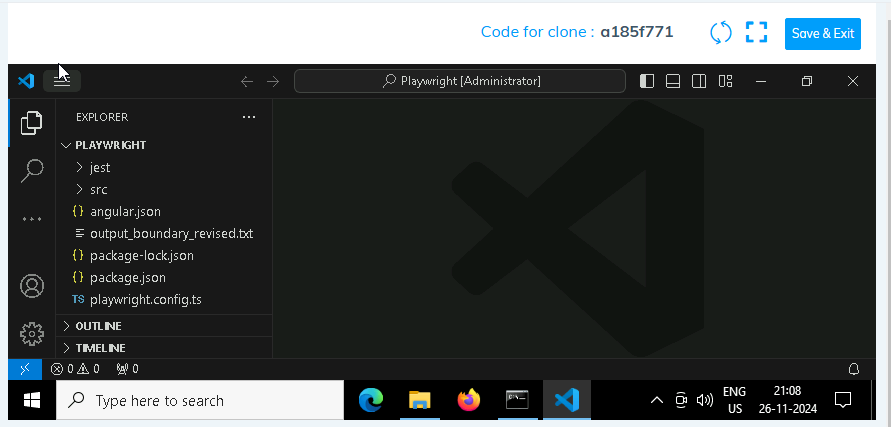
1. **Open command prompt with its location and use below command: code .**



****



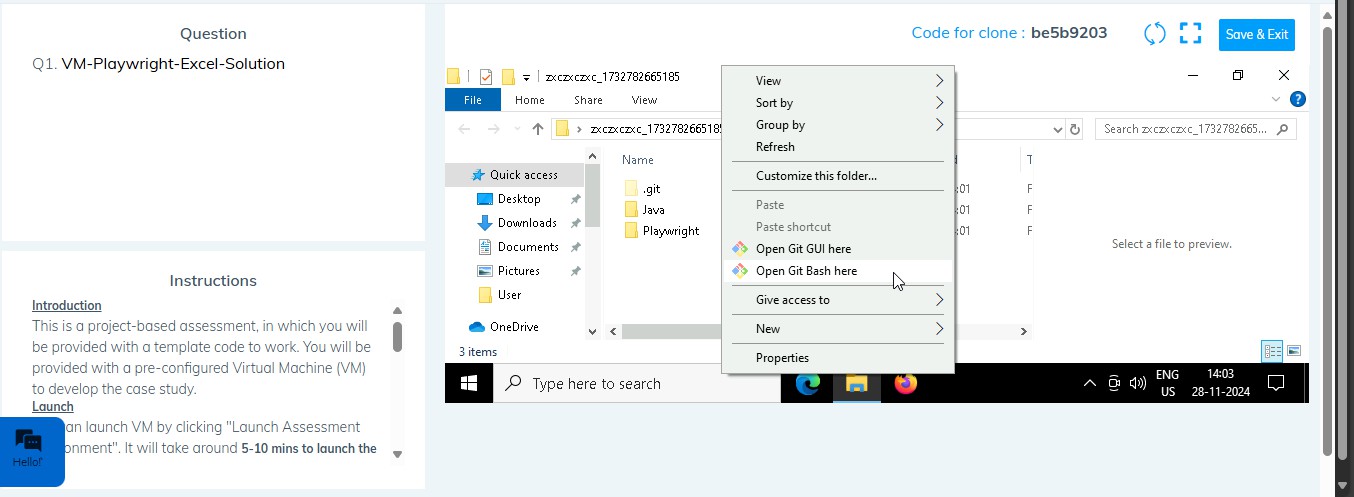
1. **Once VsCode is open. Please open the terminal in Playwright folder:**

****

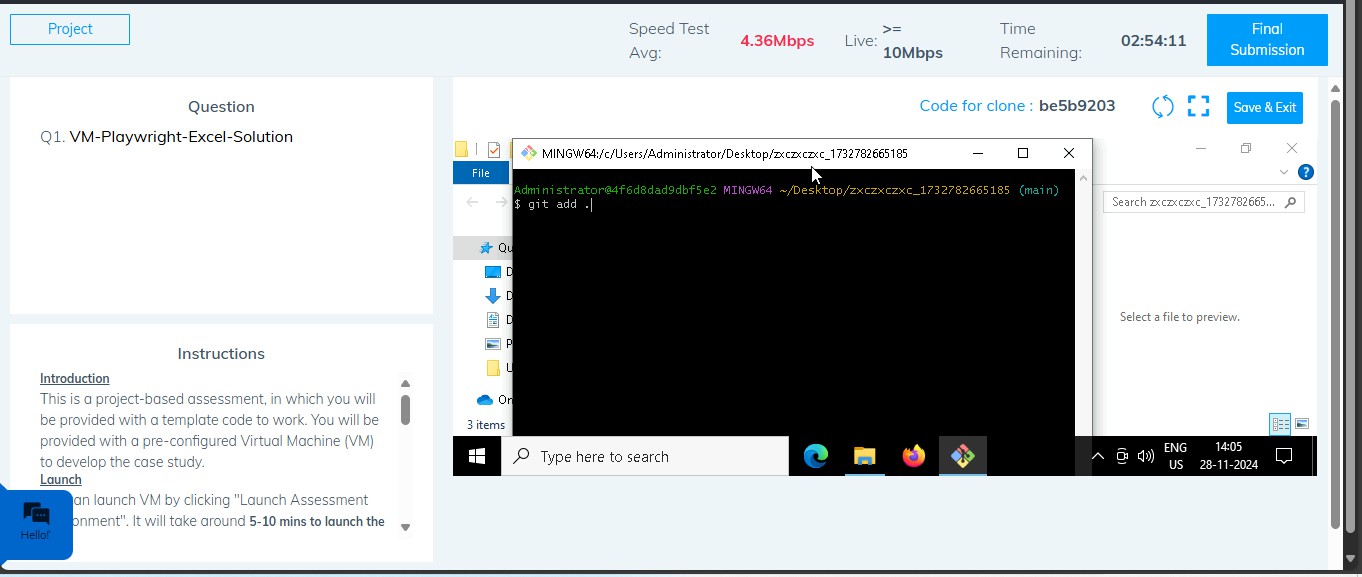
1. **Install all dependencies in the Playwright folder path using: npm install**
2. **Install playwright in the Playwright folder path: npx playwright install**
3. **Run the Tests in the Playwright folder path**:

**npx playwright test ./src/tests/PL1\_testcases/yaksha.spec.ts**

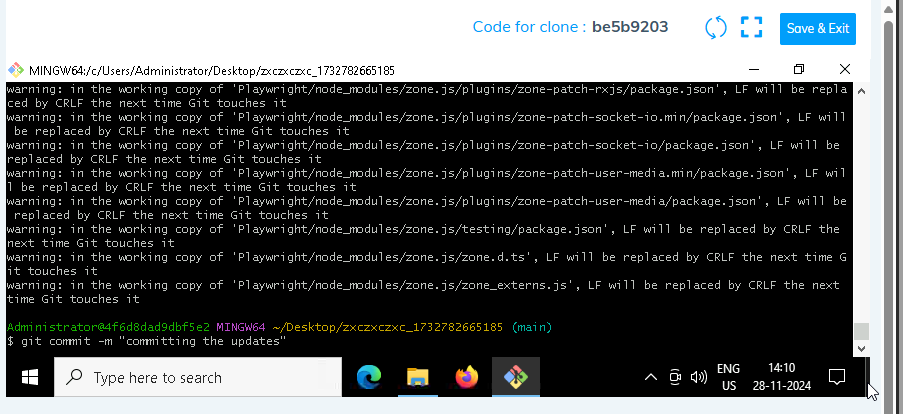
1. **Once you have executed the test cases. Now it is necessary to push your code to git. For this, please go inside the folder created on desktop with the email id you have used to login and then:**
   1. **Open gitbash**

****

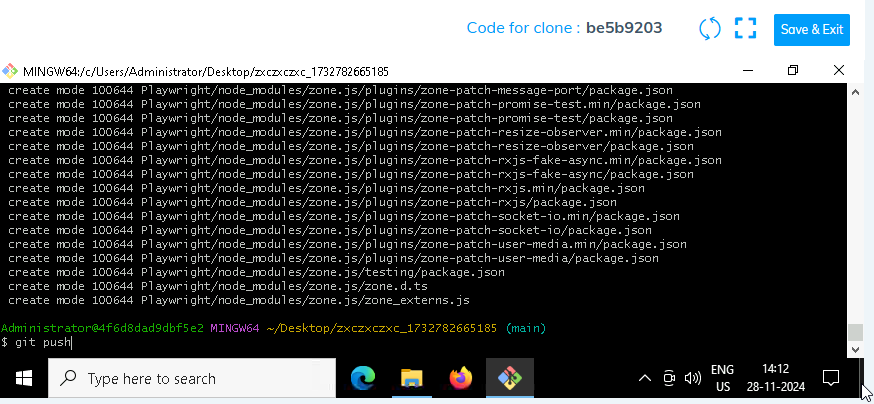
* 1. **Add all files**



* 1. **Commit the changes**



* 1. **Push the changes**

****