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 * MyInfoPage * PIMPage | 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 the 'My Info' tab Loads Succesfully | 1. Login using valid credentials  2. Click on "My Info" tab  **It must implement this login**  **functionality at first and then implement any other test case.** | **Reference path**  \src\pages\MyinfoPage  **methods**  clickMyInfoTab()  **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 My Info tab loads succesfully** |
|  |  |  | **Reference path** |  |
| 2 | Verify the 'Required' field error Message is displayed on leaving name field blank | **1. Login using valid credentials**  **2. Click on "My Info" tab**  **3. Clear the name present in the name inputbar** | \src\ pages\**MyinfoPage**  **methods**  clearAndEnterName()  **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.** | **Required' field error message should be displayed** |
|  |  |  | **Reference path** |  |
| 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 | \src\ pages\**MyinfoPage**  **Methods:- updateUniqueNAmeAndVerifyName()** | Verify the name gets updated in the top right corner |
|  |  |  | **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 on top 3 items from the sidebar ('personal details’, contact details, emergency contact') | | **Reference path** |  |
|  |  | \src\ pages\ **DependentsPage** |  |
| 4 | Verify the 'My Info' tab's subtabs have unique URL | **Methods:-**  **areMyInfoSubTabHrefsUnique()**  **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 url from the three is unique |
|  |  |  | | **Reference path** |  |
| 5 | **A new Verify Dependent 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 'child' from relationship and hit save button | | \src\ pages\ **DependentsPage**  **methods**  addDependent()  **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 presence of the child in the list** |
| 6 | **Verify new Dependent '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 input bar is not present before entering the other in relationship inputbar** |
| 7 | Verify that membership can be added from the list | **1.Login using valid credentials. 2. Click on the "My Info" tab. Navigate to the "Memberships" subtab. 3. Click on the "Add" button. Select a \*membership.**  **\*Fill the amount and save.**  **.** | | **Reference path**  \**PageObjects\Pages\MyinfoPage** **methods**: addMembership() | **The newly added membership should appear in the memberships list.** |
| 8 | **Verify that membership can be deleted from the list** | 1.Login using valid credentials. 2. Click on **"**MyInfo**"** tab. 3.Navigate to **"**Memberships**"** subtab. 4. Locate the membership to be deleted in the list. 5. Click on the **"**Delete**"** icon/button for that membership. 6. Confirm the deletion action if prompted. | | **Reference path**    \src\pages\**MyinfoPage**  **Methods: deleteMembership**  () | **The selected membership should be removed from the memberships list.** |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case No.** | **Test Case Name** | **Test Steps to be performed** | **Path & Method Used** | **Expected Result** |
|  |  | **1.Login using valid credentials. 2. Click on the "My Info" tab. 3.Navigate to the "Memberships" subtab. 4. Select the first membership from the list. 5. Click on the "Edit" icon/button. Update the membership details  6. Save the updated membership details.** | **Reference path** |  |
|  |  | \src\ pages\**MyinfoPage** |  |
|  |  | **methods** |  |
| 9 | Verify that membership can be edited from the list | editMembership()  **You can use highlightElement method present in CommonMethods file to highlight the element before performing any action on it. It takes locator as** | The updated membership details should be reflected correctly in the memberships list after saving. |
|  |  | 1.Login using valid credentials. 2. Click on **"**MyInfo**"** tab. 3.Navigate to **"**Memberships**"** subtab. 4. Click on the **"**AddAttachment**"** button/icon. 5. Upload a valid file from the system. 6. Enter a comment in the comment field. 7. Save the attachment. | **a parameter.**  **Reference path**  \src\pages\**MyinfoPage** |  |
| 10 | Verify Primary Color of corporate branding could be changed |  | **methods**  addmembershipAttach() | The uploaded attachment along with the comment should appear in the attachments list under the Memberships subtab. |
|  |  | **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 Immigrant assign immigration record Add Functionality | 1.Login using valid credentials. 2. Click on **"**MyInfo**"** tab. 3.Navigate to **"**Immigrations**"** subtab. 3. Click on the **"**Add**"** button. 4. Fill in all required immigration record details. 5. Submit the form.  **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: addImegration()  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 newly added immigration record should appear in the immigration records list with all details correctly displayed. |
| 12 | Verify the Subtabs css change after click on the subtab | 1. Navigate to URL with Valid Credentials  2. Navigate to "admin" tab  3. Click on the subtab  **Note: - If you cannot find the locator for “Proceed” button, please try to keep the page at 75%-80%.** | **Reference path**  \**src\pages\AdminPage**  Methods: **hoverAdmin**()  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 subtab appearance changes when clicked upon subtab. |
| 13 | Verify the Immigrant assign immigration record Delete Functionality | 1. Login using valid credentials  2. Click on "My Info" tab  3. Navigate to immigration subtab  4. Click the 'Delete' button from desired record **Note**: **- If you cannot find the locator for “Proceed” button, please try to keep the page at 75%-80%.** | **Reference path**  \**src\pages\PIMPage**  Methods: **EmpIdError**()  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 record gets deleted Successfully |
| 14 | Verify the Immigrant assign immigration record Delete selected Functionality | 1. Login using valid credentials  2. Click on "My Info" tab  3. Navigate to 'Dependents' subtab  4. Select top 3 from the list  5. Click delete selected and 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: checkboxDeleteFunction()**  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. | Verify the selected record gets deleted Successfully |
| 15 | Verify the My Info's Qualification tab Loads Successfully | 1. Login using valid credentials  2. Click on "My Info" tab  3. Click 'Qualification' from the sidebar  **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: -qualificationTab()**  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 URL with 'viewQualifications' Loads Successfully |

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**

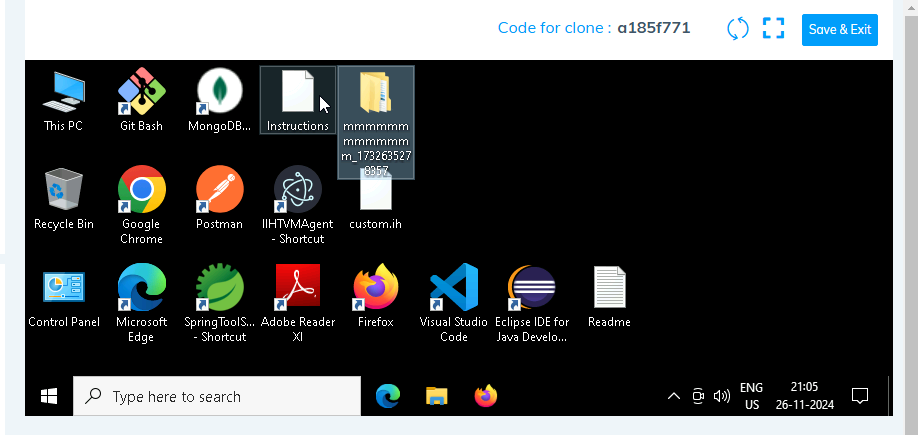
# Associates should have written clean code that is readable.

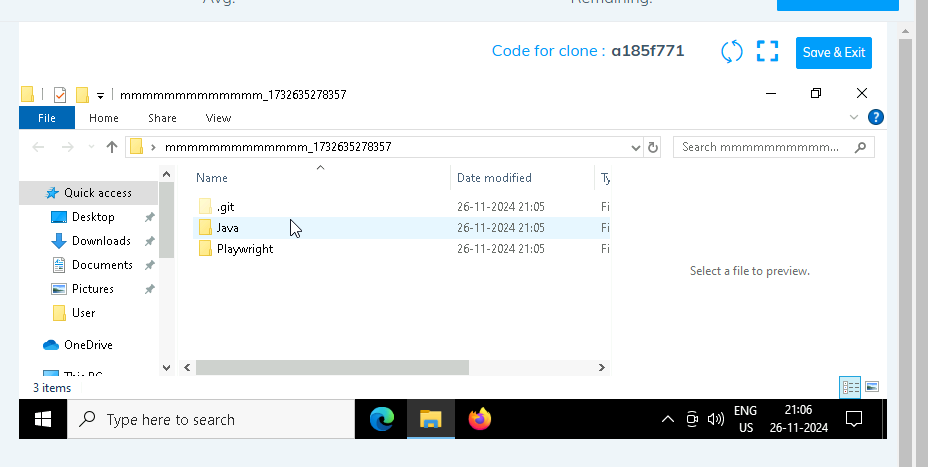
* + 1. Associates need to follow SOLID programming principles.

**Execution Steps:**

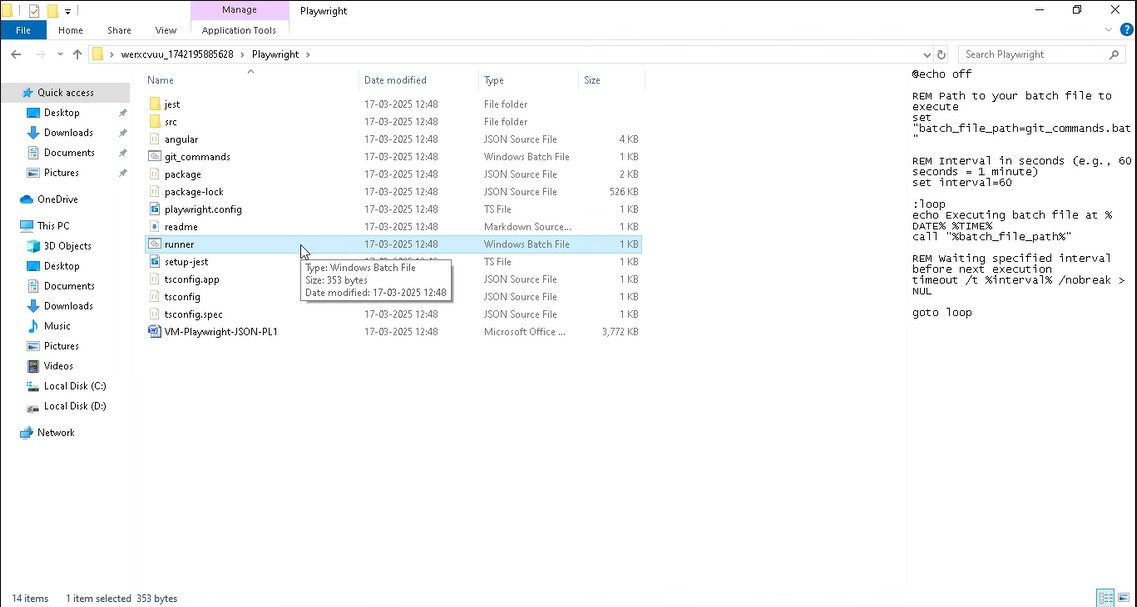
**Steps for Execution:**

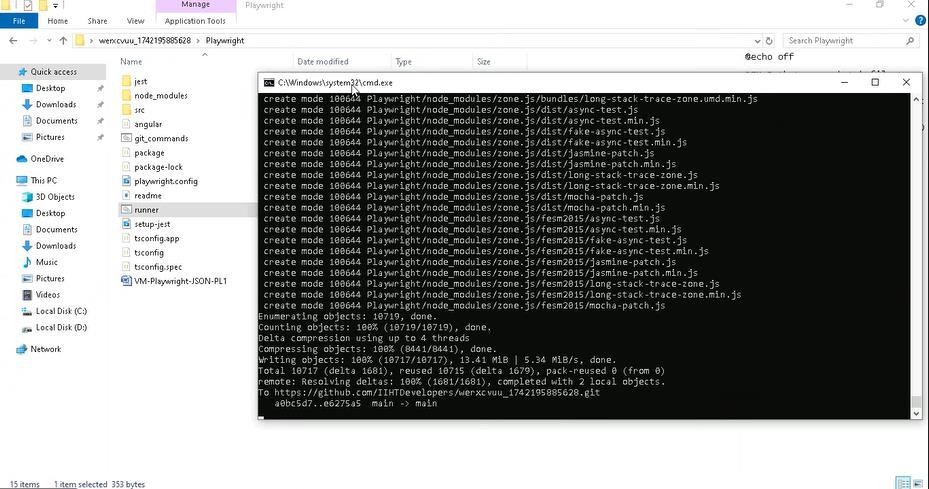
1. **Please open the folder created on desktop with the email name you used to login.**

****

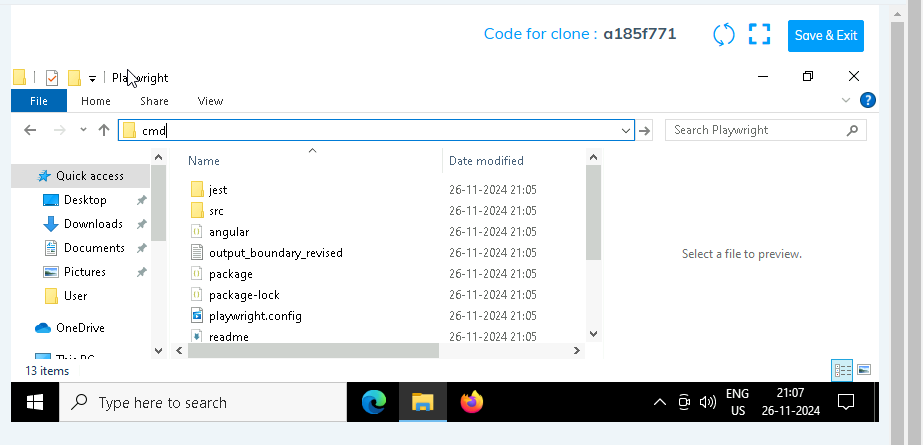


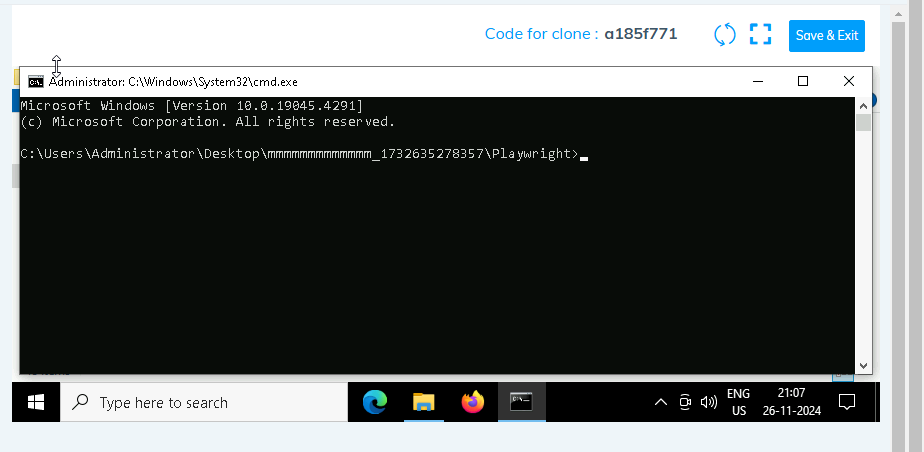
1. **Go into the Playwright folder and execute this “runner” file. This will keep pushing the code at regular intervals**

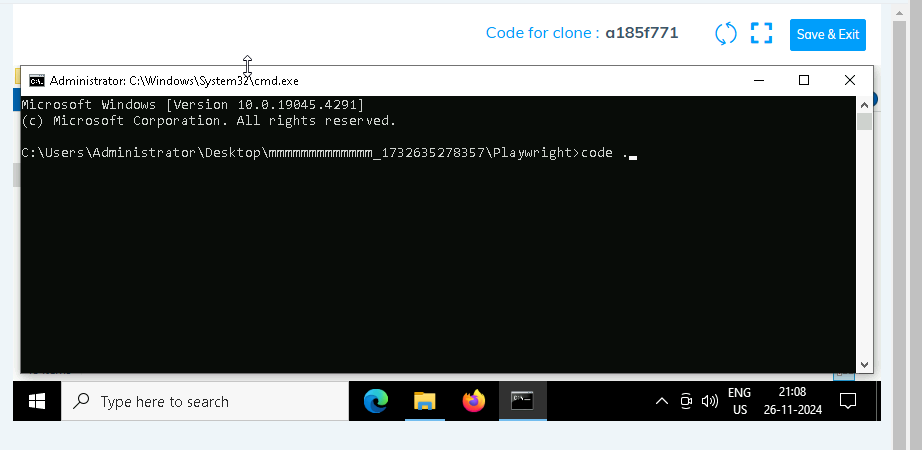


****

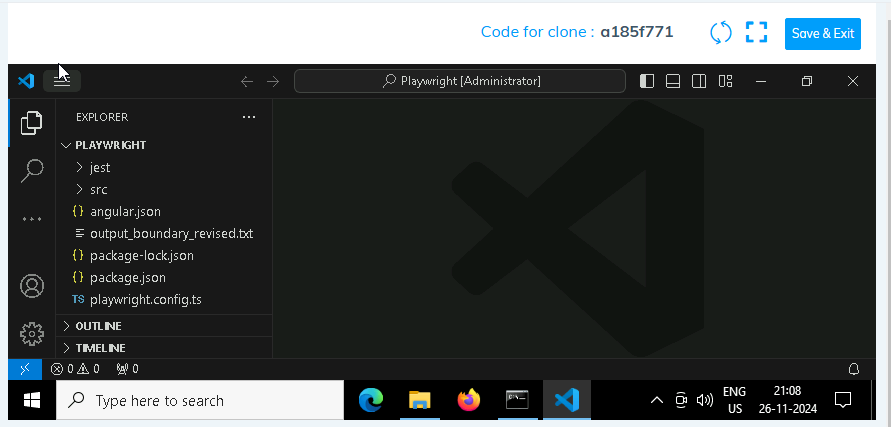
1. **Open command prompt with its location and use below command: code .**



****



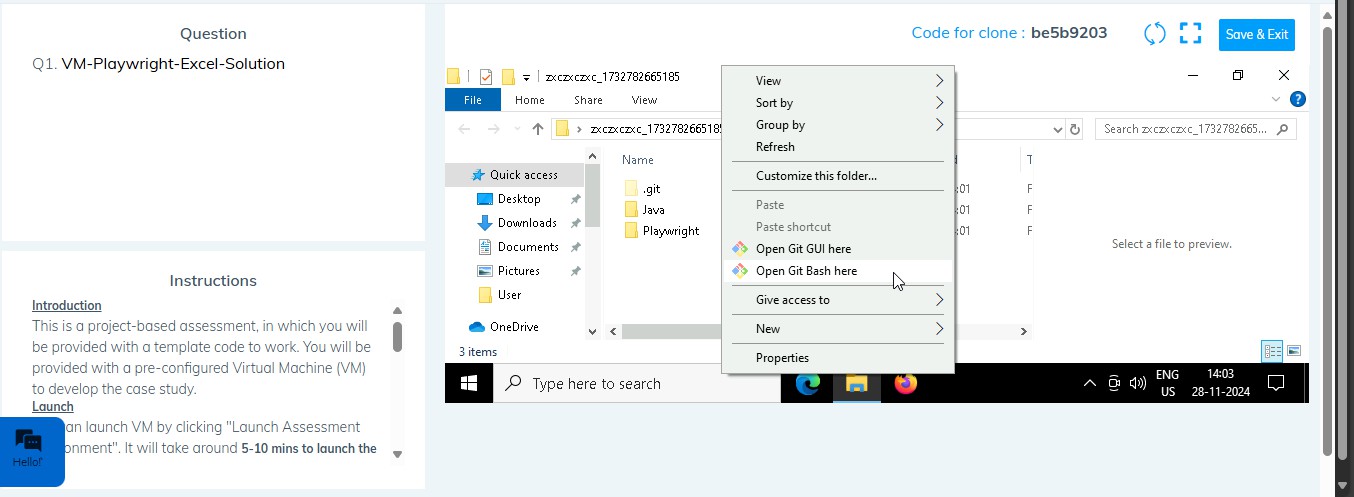
1. **Once VsCode is open. Please open the terminal in Playwright folder:**

****

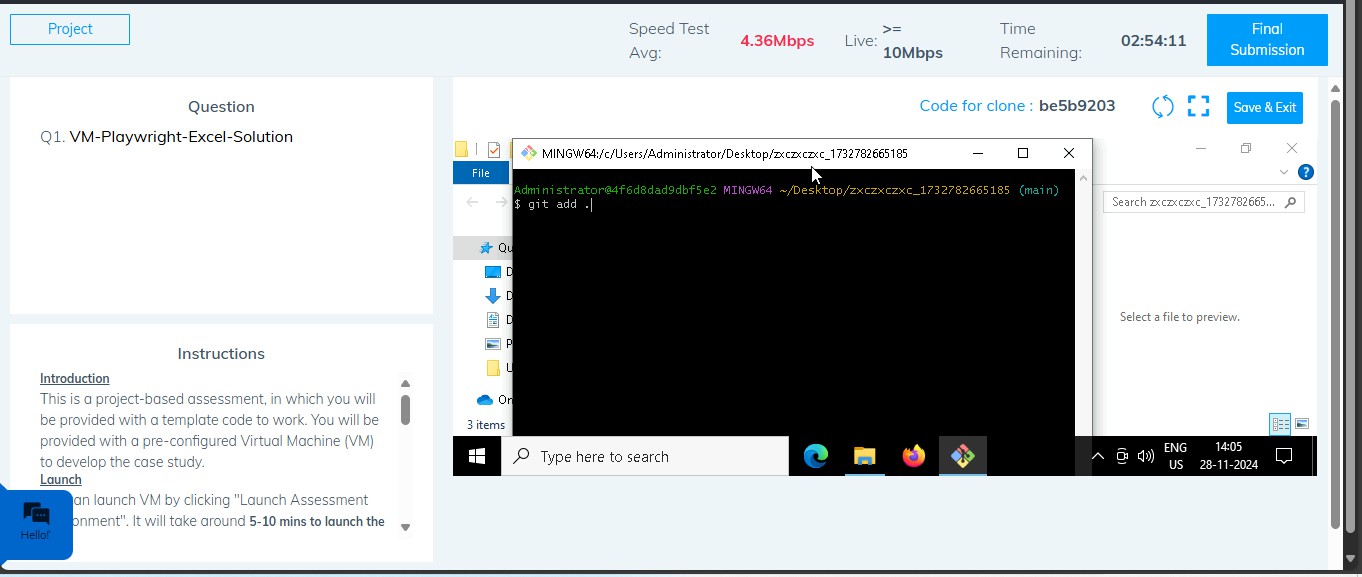
1. **Install all dependencies in the Playwright folder path using: npm install**
2. **Install playwright in the Playwright folder path: npx playwright install**
3. **Run the Tests in the Playwright folder path**:

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

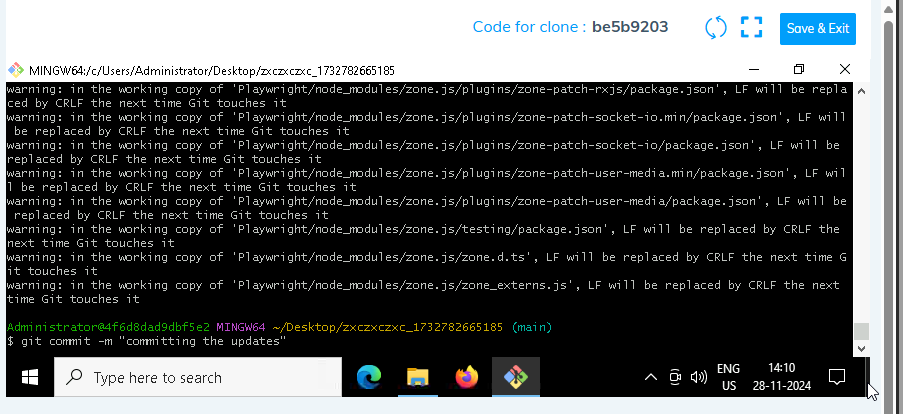
1. **Once you have executed the test cases. Now it is necessary to push your code to git. For this, please go inside the folder created on desktop with the email id you have used to login and then:**
   1. **Open gitbash**

****

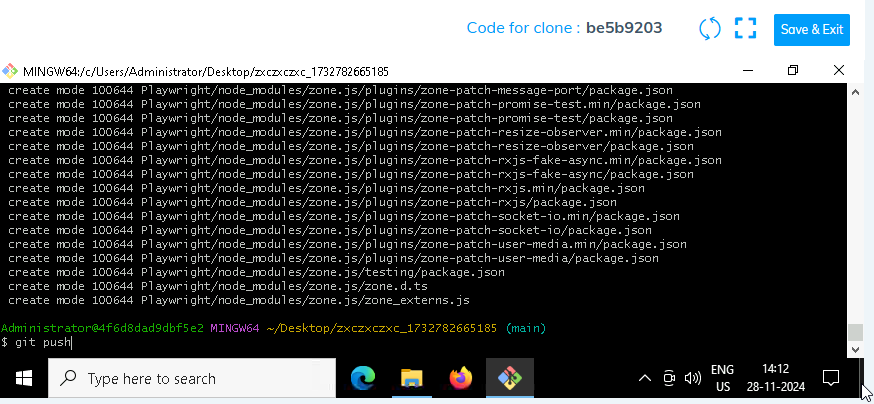
* 1. **Add all files**



* 1. **Commit the changes**



* 1. **Push the changes**

****