

# YAKSHA HEALTH APP WITH JAVASCRIPT AND CYPRESS

OrangeHRM automation using

## 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: Cypress (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 cypress+javascript

**You will be given few Json files in fixtures folder like doctor.json, login.json, maternity.json, medicalRecord.json, patientName.json, pharmacy.json, radiology.json, settings.json and subStore.json.**

Path	File	Description
\fixtures	AddUserForm.Json HolidayForm.json LoginData.json skyImage.jpg	1. Contains data to read from json file.

OrangeHRM automation using

PageObjects\Pages	<ul style="list-style-type: none"> <li>• AdminPage</li> <li>• HomePage</li> <li>• LoginProfilePage</li> <li>• ProfilePage</li> <li>• LeavePage</li> </ul>	<ol style="list-style-type: none"> <li>1. All core activities to be performed here.</li> <li>2. The comments associated with each templated method here describe the expectation.</li> <li>3. Declare any variable/object you need to share data/status between different methods.</li> </ol> <p>Do not modify the signature of methods declared here.</p>
-------------------	---	--

**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 Log Out Button Working	<ol style="list-style-type: none"> <li>1. Navigate to the profile dropdown using LogoutFuntionCheck()</li> <li>2. Click on the "Logout" button from the dropdown</li> <li>3. Verify logout completed with verifyLogoutCompleted()</li> </ol> <p><b>Images must exist in cypress/fixtures</b> <b>User must be logged in.</b></p>	<p><b>Located in</b> PageObjects/Pages/HomePage.js, spec.js</p> <p><b>Methods:</b> LogoutFuntionCheck(), verifyLogoutCompleted()</p>	User is redirected to the login page, confirming logout was successful.
2	Verify New Holidays Could Be Created	<ol style="list-style-type: none"> <li>1. Navigate to the Leaves tab and open the Holiday section using leaveCreation()</li> <li>2. Fill the form using fixture data from holidayForm.json</li> <li>3. Click on Save.</li> <li>4. Use verifyHoldaiyCreaded() to confirm it was added to the table</li> </ol> <p><b>User must be logged in with Admin privileges.</b></p>	<p><b>Reference path</b> PageObjects/Pages/LeavePage.js:</p> <p><b>Methods:</b> leaveCreation(), verifyHoldaiyCreaded()</p>	Edit user form should appear for the selected record

Test Case No.	Test Case Name	Test Steps to be performed	Path & Method Used	Expected Result
3	Verify Password Mismatch Error display	<ol style="list-style-type: none"> <li>1. Navigate to Admin tab using passwordMismatchError()</li> <li>2. Fill Add Admin User form with mismatched passwords (data from AddUserForm.json)</li> <li>3. Attempt to save.</li> <li>4. Use verifyPasswordMismatchError() to check error message</li> </ol> <p><b>User must be logged in with Admin access.</b></p>	<p><b>Reference path</b></p> <p>PageObjects/Pages/AdminPage.js:</p> <p><b>Methods</b></p> <p>passwordMismatchError(), verifyPasswordMismatchError()</p>	Error message " <b>Passwords do not match</b> " should be displayed.
4	Verify Admin can add record	<ol style="list-style-type: none"> <li>1. Navigate to Admin tab using addAdminUserForm()</li> <li>2. Fill in all the user details (role, employee name, status, username, password, confirm password) from AddUserForm.json</li> <li>3. Click Save</li> <li>4. Use verifyAddAdmin() to verify success</li> </ol> <p><b>User must be logged in and on the home page</b></p>	<p><b>Reference path</b></p> <p>PageObjects/Pages/AdminPage.js</p> <p><b>Methods</b></p> <p>addAdminUserForm() verifyAddAdmin()</p>	Admin record is saved successfully with toast confirmation
5	Verify User should be deleted from the table	<ol style="list-style-type: none"> <li>1. Navigate to Admin tab using deleteUserFromTable()</li> <li>2. Sort User Role column in descending order to bring non-admin users to top</li> <li>3. Click delete icon for first listed user</li> <li>4. Click "Yes, Delete" to confirm</li> <li>5. Use verifyDeleteUser() to check the deleted user no longer appears.</li> </ol> <p><b>User must be logged in with Admin privileges.</b></p>	<p><b>Reference path</b></p> <p>PageObjects/Pages/AdminPage.js</p> <p><b>Methods</b></p> <p>deleteUserFromTable() verifyDeleteUser()</p>	Selected user is successfully deleted and removed from table

Test Case No.	Test Case Name	Test Steps to be performed	Path & Method Used	Expected Result
6	Verify Image Could Be Uploaded as Profile Pic	<ol style="list-style-type: none"> <li>1. Navigate to <b>My Info</b> section</li> <li>2. Open the profile picture section.</li> <li>3. Upload image from fixtures.</li> <li>4. Click Save on Save Button</li> <li>5. Verify updated image using verifyProfilePictureUpdated().</li> </ol> <p><b>Images must exist in cypress/fixtures</b> <b>User must be logged in</b></p>	<p><b>Located in</b> PageObjects/Pages/ProfilePage.js:</p> <p><b>Methods :</b></p> <pre>uploadProfilePicture('skyimage.jpg'); verifyProfilePictureUpdated();</pre>	Profile image is successfully updated and visible
7	Verify Admin Can Edit Record	<ol style="list-style-type: none"> <li>1. Navigate to <b>Admin Page</b> using</li> <li>2. Click <b>Edit</b> button on the first listed user.</li> <li>3. Verify edit form appears using verifyEditFormAppears()</li> </ol> <p><b>At least one user should exist in the user list</b> <b>User must be logged in with Admin privileges.</b></p>	<p><b>Reference path</b> PageObjects/Pages/AdminPage.js:</p> <p><b>Methods:</b></p> <pre>verifyEditFormAppears(); verifyUserEditFormAppears</pre>	Edit user form should appear for the selected record
8	Verify Admin Can Sort the Records	<ol style="list-style-type: none"> <li>1. Navigate to <b>Admin Page</b>.</li> <li>2. Click on the <b>Username</b> column</li> <li>3. Validate the usernames are sorted</li> </ol> <p><b>User must be logged in with Admin access At least two records with different usernames must exist</b></p>	<p><b>Reference path</b> PageObjects/Pages/AdminPage.js:</p> <p><b>Methods</b></p> <pre>goToAdminAndSortByUsername() verifyUsernamesAreSortedAsc()</pre>	The list of usernames should be sorted in ascending order after clicking the sort button.
9	Verify Upgrade Page Loads Correctly After Clicking the Button	<ol style="list-style-type: none"> <li>1. Navigate to the <b>Home Page</b></li> <li>2. Locate and click the <b>Upgrade</b> button/link.</li> <li>3. Validate its href and target attributes using verifyUpgradeLinkAttributes()</li> </ol> <p><b>User must be logged in and on the home page</b></p>	<p><b>Reference path</b> PageObjects/Pages/HomePage.js</p> <p><b>Methods</b></p> <pre>verifyUpgradeLinkAttributes()</pre>	<p>The Upgrade button should:</p> <ul style="list-style-type: none"> <li>- Have a valid href to upgrade URL</li> <li>- Open in a new tab (target="_blank")</li> </ul>

Test Case No.	Test Case Name	Test Steps to be performed	Path & Method Used	Expected Result
10	Verify on Hover Shows Help Text	1. Navigate to the <b>Admin Page</b> 2. Locate the Help icon. 3. Trigger hover Verify title="Help" or tooltip text appears	<b>Reference path</b> PageObjects/Pages/AdminPage.js  <b>Methods</b> visitAdminForTolltip() verifyHoverWork	The Help icon should display tooltip or attribute containing the text " <b>Help</b> ".

Learners will gain experience in building strongly-typed applications using React.js and managing data flow with **JavaScript**. They'll learn how to define interfaces, use types for error prevention, and improve code maintainability.

With **Cypress**, learners will learn to write and execute automated tests for the <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

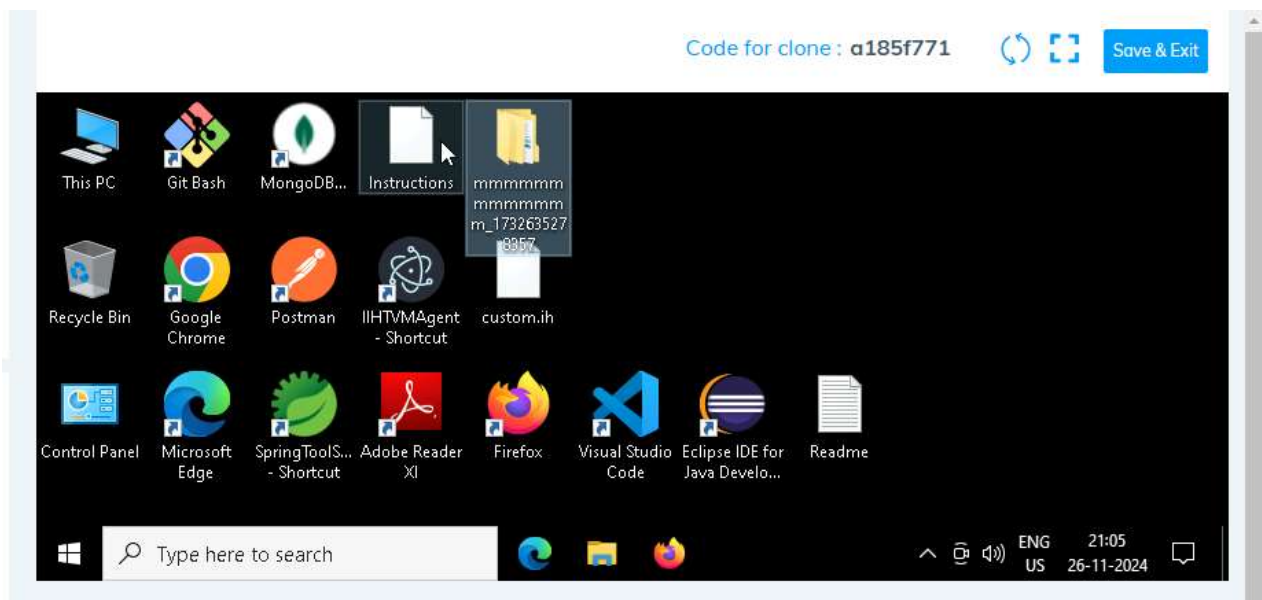
### 1.1 CODE QUALITY/OPTIMIZATIONS

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

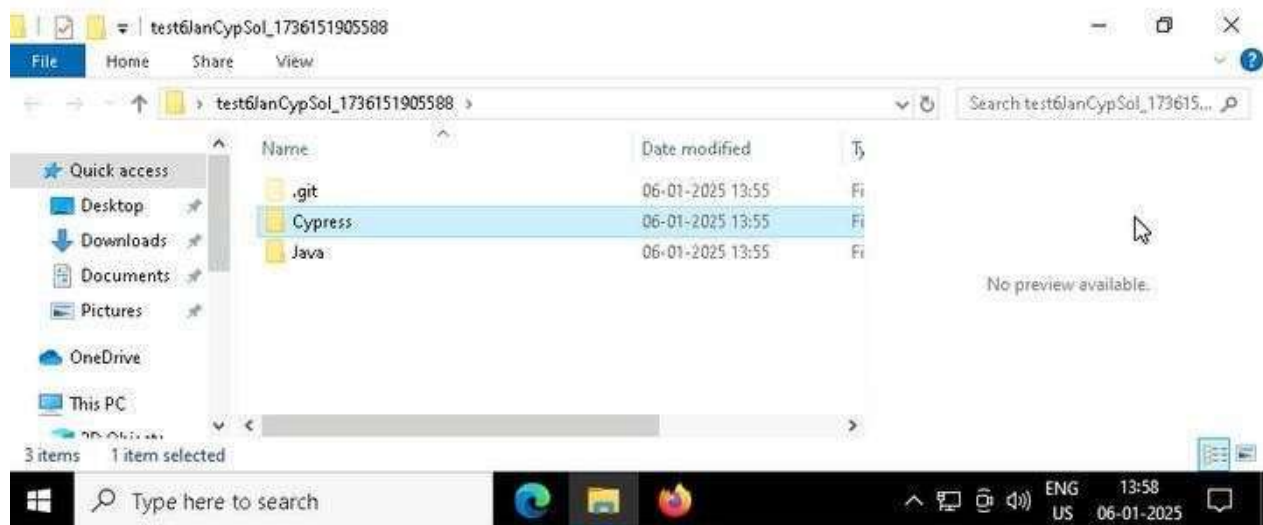
**Execution Steps:**

**Steps for Execution:**

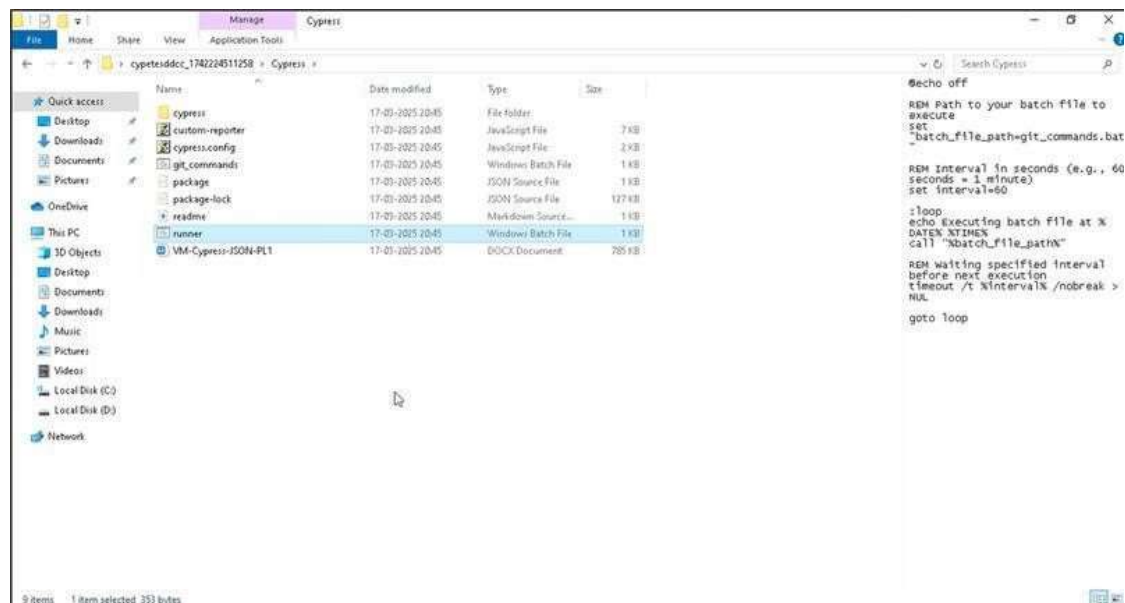
1. Please open the folder created on desktop with the email name you used to login.



OrangeHRM automation using

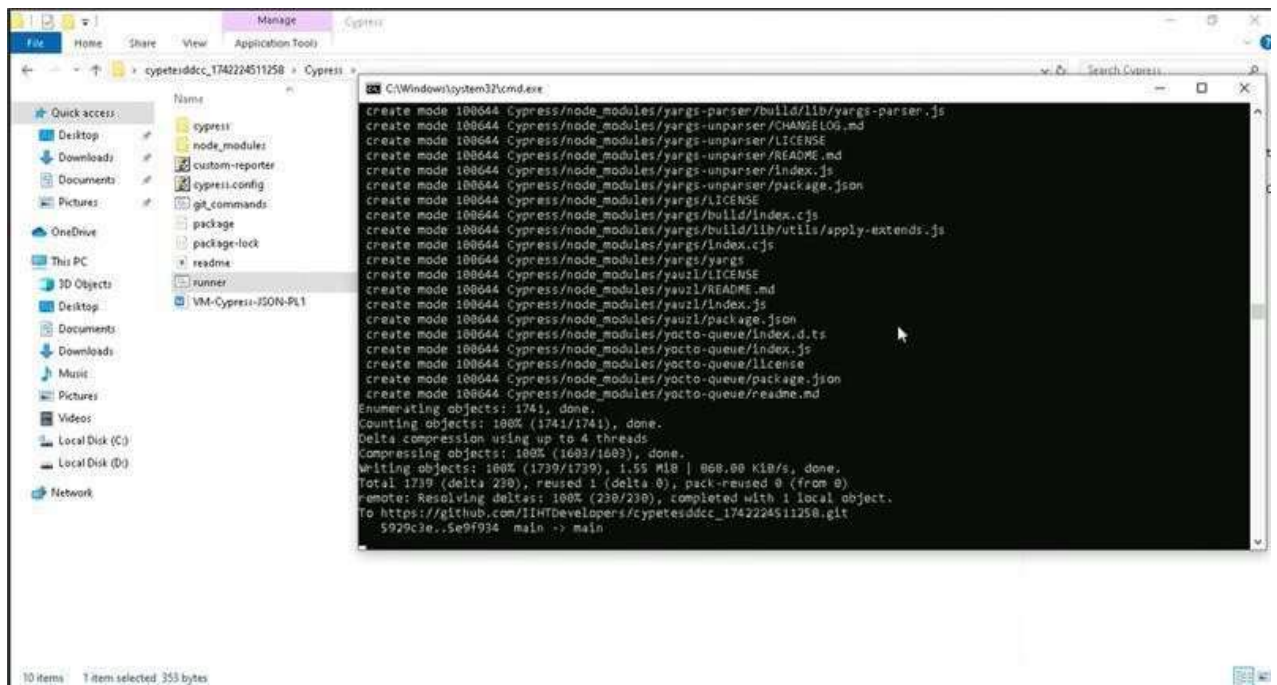


- Go into the Cypress folder and execute this “runner” file. This will keep pushing the code at regular intervals.

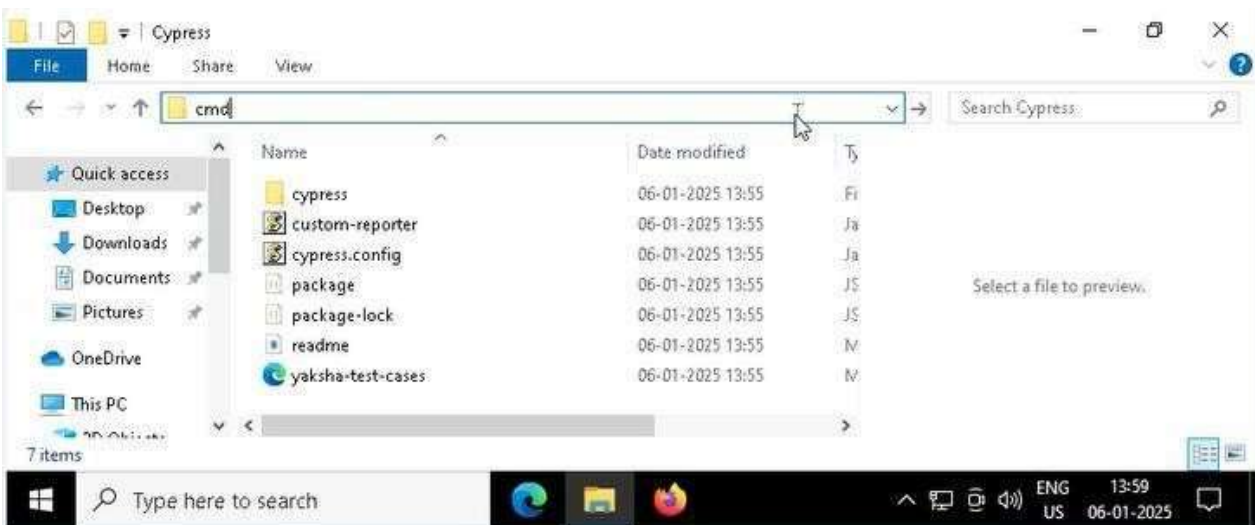


OrangeHRM automation using

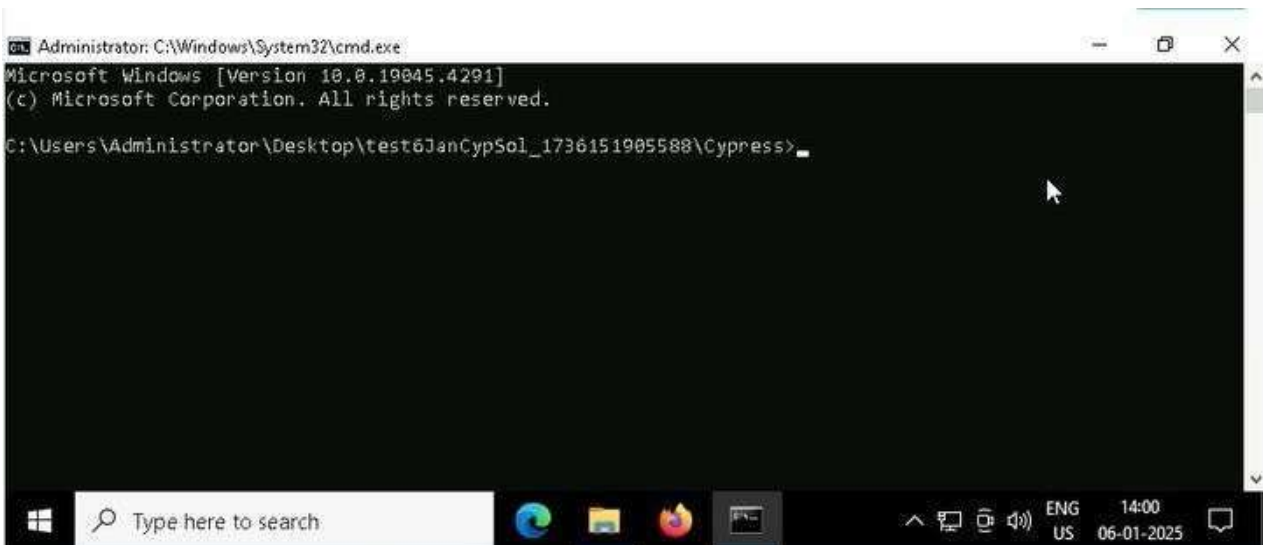




### 3. Open command prompt with its location using “cmd” in search bar

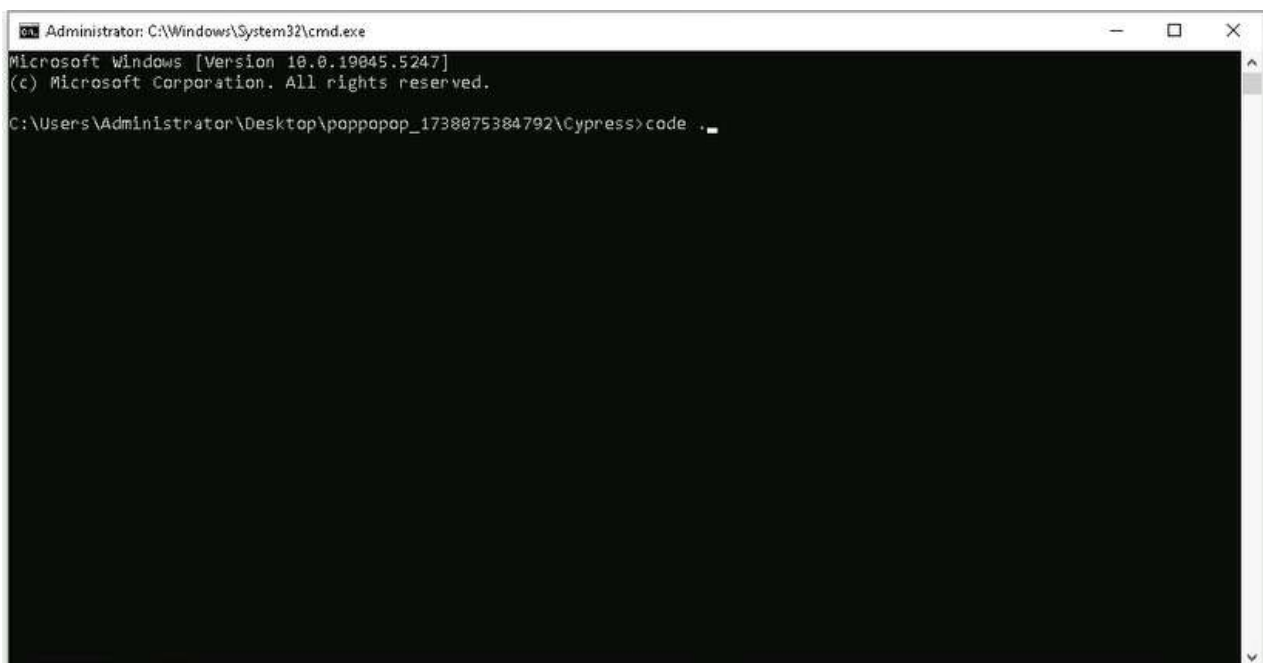


OrangeHRM automation using



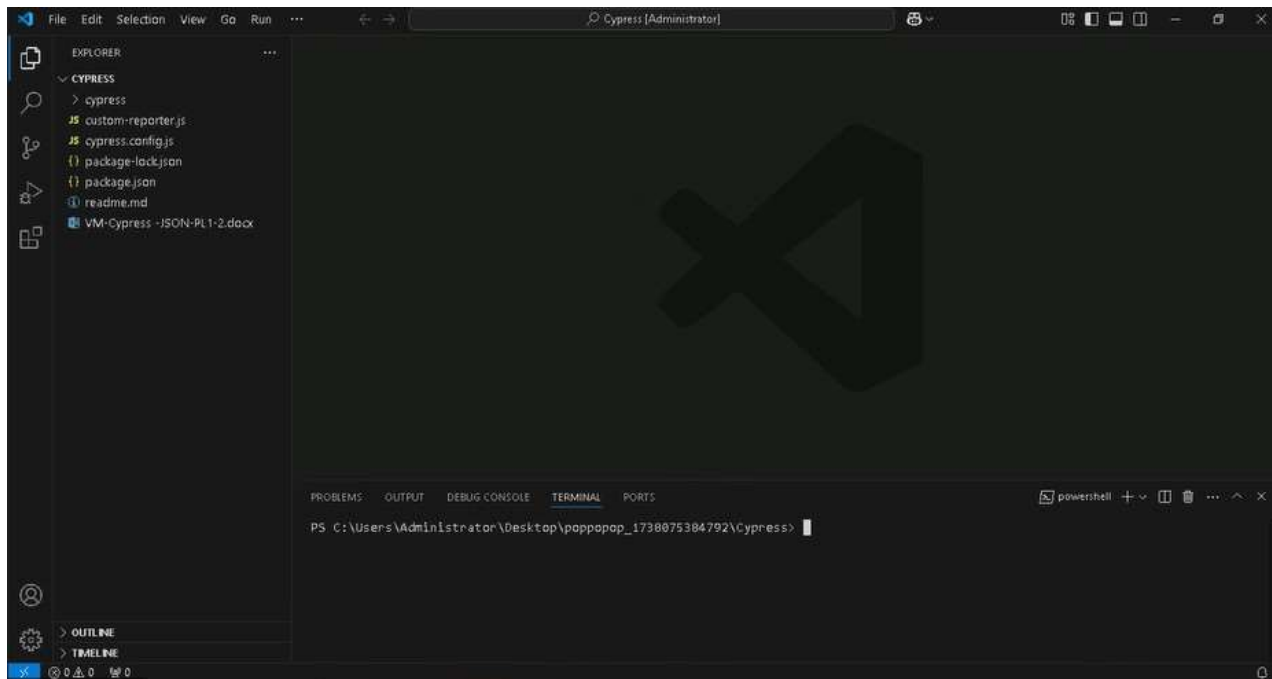
4. Open VS Code through cmd using below command:

`code .`



OrangeHRM automation using

**5. Once VsCode is open. Please open the terminal in Cypress folder:**



**1. Install all dependencies in the Cypress folder path using:**

**npm install**

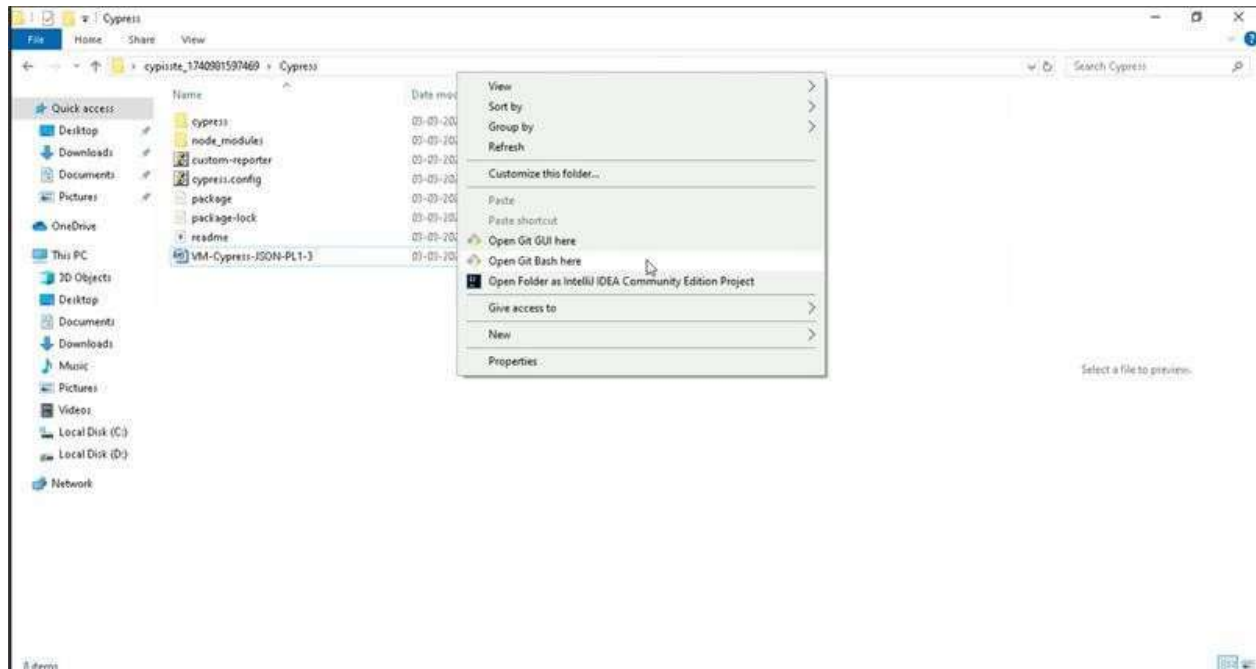
**2. Run the following command to open the interactive Cypress Test Runner in the Cypress folder path:**

**npx cypress open**

**3. Run the following command to run test cases in headless mode in the Cypress folder path:**

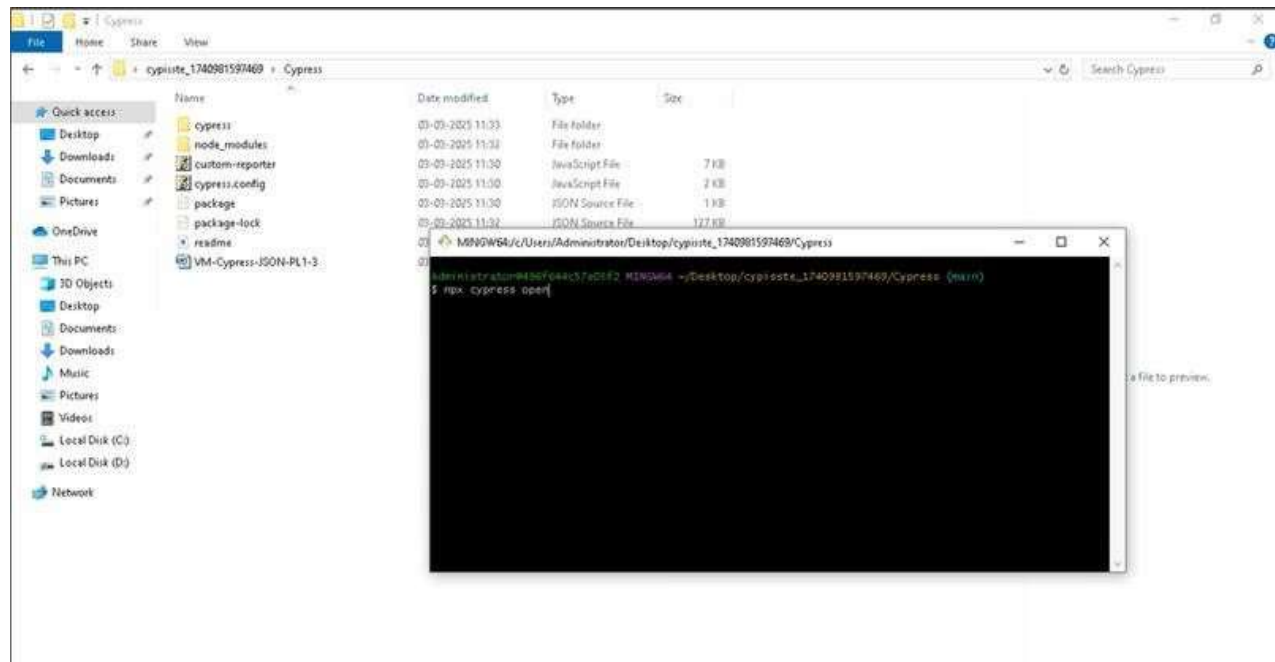
**npx cypress run**

If you face any issue in executing above commands i.e `npx cypress open` and `npx cypress run` in cmd. Please try to run these in git bash from cypress folder as:



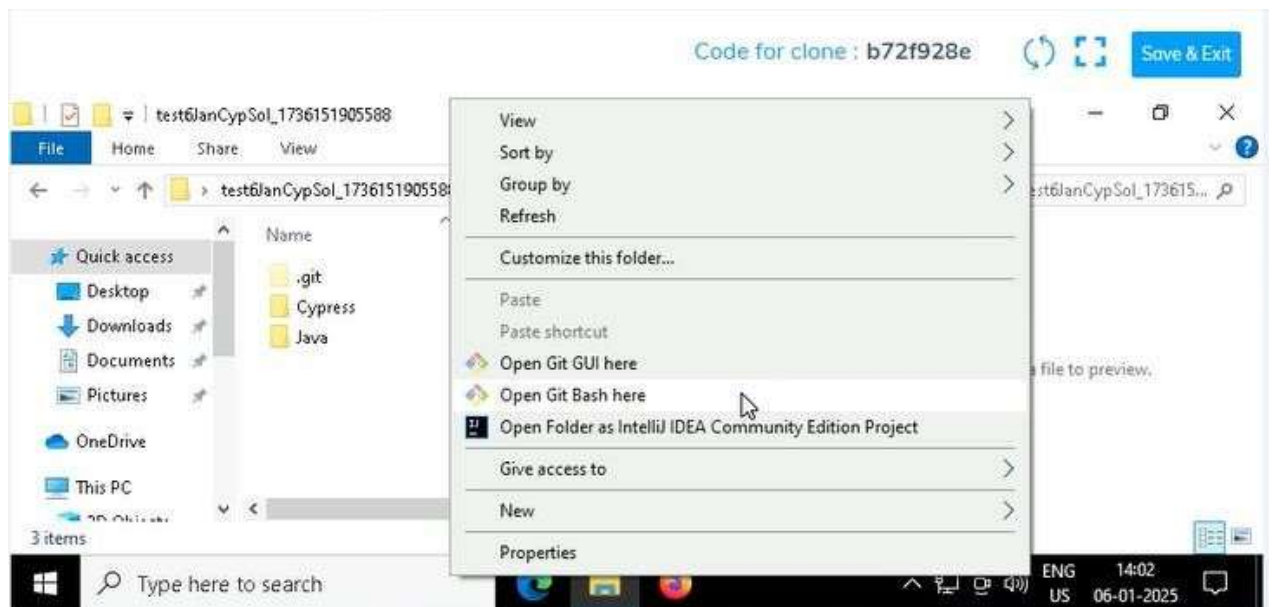
OrangeHRM automation using

and then execute commands like:



4. 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



OrangeHRM automation using

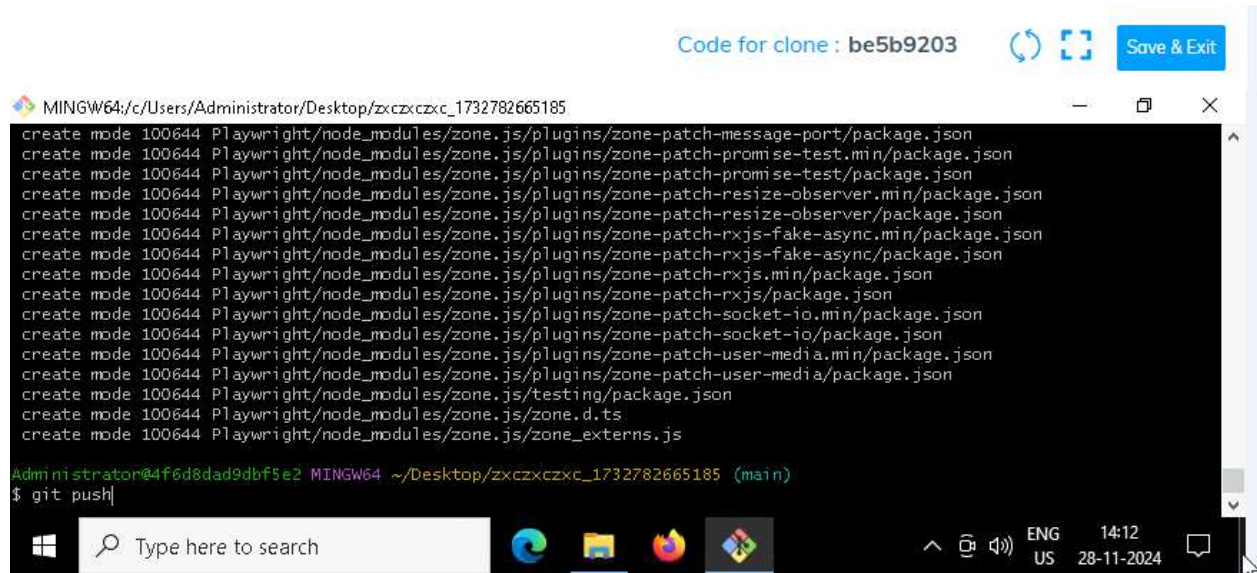
## 2. Add all files

The screenshot shows the OrangeHRM interface. On the left, under the 'Question' tab, is 'Q1. VM-Playwright-Excel-Solution'. Below it, under 'Instructions', is an 'Introduction' section stating: 'This is a project-based assessment, in which you will be provided with a template code to work. You will be provided with a pre-configured Virtual Machine (VM) to develop the case study.' There is also a 'Launch' button with a VM icon and text: 'You can launch VM by clicking "Launch Assessment Environment". It will take around 5-10 mins to launch the VM.' On the right, there's a 'Speed Test' section showing 'Avg: 4.36Mbps' and 'Live: >= 10Mbps'. Below that, a 'Code for clone : be5b9203' is displayed with a refresh icon and a 'Save & Exit' button. A terminal window is open in the center, showing the command 'git add .' being executed. The terminal title bar indicates the path 'MINGW64/c/Users/Administrator/Desktop/zxczxc\_1732782665185'. The Windows taskbar at the bottom shows the search bar and system clock '14:05 28-11-2024'.

## 3. Commit the changes

This screenshot shows the terminal window from the previous step, now displaying the output of the 'git add .' command. Multiple warnings are shown, indicating that line endings (LF) will be replaced by CRLF in various files within the 'Playwright/node\_modules/zone.js/plugins/' directory. The terminal title bar remains the same. The 'Code for clone : be5b9203' and 'Save & Exit' button are still visible at the top right. The terminal shows the command '\$ git commit -m "committing the updates"' being entered. The Windows taskbar at the bottom shows the search bar and system clock '14:10 28-11-2024'.

#### 4. Push the changes



The screenshot shows a Windows terminal window with a dark background. At the top, there is a blue header bar with the text "Code for clone : be5b9203" and a "Save & Exit" button. Below the header, the terminal displays the output of a git push command. The output shows a series of "create mode" messages for various files in the Playwright/node\_modules/zone.js/plugins directory. The files include zone-patch-message-port/package.json, zone-patch-promise-test.min/package.json, zone-patch-promise-test/package.json, zone-patch-resize-observer.min/package.json, zone-patch-resize-observer/package.json, zone-patch-rxjs-fake-async.min/package.json, zone-patch-rxjs-fake-async/package.json, zone-patch-rxjs.min/package.json, zone-patch-rxjs/package.json, zone-patch-socket-io.min/package.json, zone-patch-socket-io/package.json, zone-patch-user-media.min/package.json, zone-patch-user-media/package.json, zone-testing/package.json, zone.d.ts, and zone\_externs.js. The terminal prompt shows the user is Administrator@4f6d8dad9dbf5e2 in a MINGW64 shell, and the current directory is ~/Desktop/zxczxcxc\_1732782665185 (main). The user has entered the command \$ git push. The Windows taskbar is visible at the bottom, showing the Start button, a search bar, and several application icons. The system clock shows 14:12 on 28-11-2024.

```
Code for clone : be5b9203 Save & Exit
MINGW64~/c/Users/Administrator/Desktop/zxczxcxc_1732782665185
create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-message-port/package.json
create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-promise-test.min/package.json
create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-promise-test/package.json
create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-resize-observer.min/package.json
create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-resize-observer/package.json
create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-rxjs-fake-async.min/package.json
create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-rxjs-fake-async/package.json
create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-rxjs.min/package.json
create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-rxjs/package.json
create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-socket-io.min/package.json
create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-socket-io/package.json
create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-user-media.min/package.json
create mode 100644 Playwright/node_modules/zone.js/plugins/zone-patch-user-media/package.json
create mode 100644 Playwright/node_modules/zone.js/testing/package.json
create mode 100644 Playwright/node_modules/zone.js/zone.d.ts
create mode 100644 Playwright/node_modules/zone.js/zone_externs.js
Administrator@4f6d8dad9dbf5e2 MINGW64 ~/Desktop/zxczxcxc_1732782665185 (main)
$ git push
```

OrangeHRM automation using