

# YAKSHA HEALTH APP WITH JAVASCRIPT AND CYPRESS PL1-4

## 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 AddUserForm.json, HolidayForm.json, LoginData.json and SkylImage.jpg.**

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

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> </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()</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> </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()</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> </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()</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> </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()</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> </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()</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	1.Navigate to <b>My Info</b> section 2.Open the profile picture section. 3.Upload image from fixtures. 4.Click Save on Save Button  <b>Images must exist in cypress/fixtures</b> <b>User must be logged in</b>	<b>Located in</b> PageObjects/Pages/ProfilePage.js:  <b>Methods :</b>  uploadProfilePicture('skyimage.jpg');	Profile image is successfully updated and visible
7	Verify Admin Can Edit Record	1. Navigate to <b>Admin Page</b> using 2. Click <b>Edit</b> button on the first listed user.  <b>At least one user should exist in the user list</b> <b>User must be logged in with Admin privileges.</b>	<b>Reference path</b> PageObjects/Pages/AdminPage.js:  <b>Methods:</b>  editUserFormAppear()	Edit user form should appear for the selected record
8	Verify Admin Can Sort the Records	1. Navigate to <b>Admin Page</b> . 2. Click on the <b>Username</b> column 3. Validate the usernames are sorted  <b>User must be logged in with Admin access At least two records with different usernames must exist</b>	<b>Reference path</b> PageObjects/Pages/AdminPage.js:  <b>Methods</b>  goToAdminAndSortByUsername()	The list of usernames Should be sorted in ascending order after clicking the sort button.
9	Verify new tab opens on clicking the upgrade button	1. Navigate to the <b>Home Page</b> 2. Locate and click the <b>Upgrade</b> button.  <b>User must be logged in and on the home page</b>	<b>Reference path</b> PageObjects/Pages/AdminPage.js  <b>Methods</b>  NavigateToAdminPage()	The Upgrade button should: - Have a valid href to upgrade URL - Open in a new tab (target="_blank")

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()	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

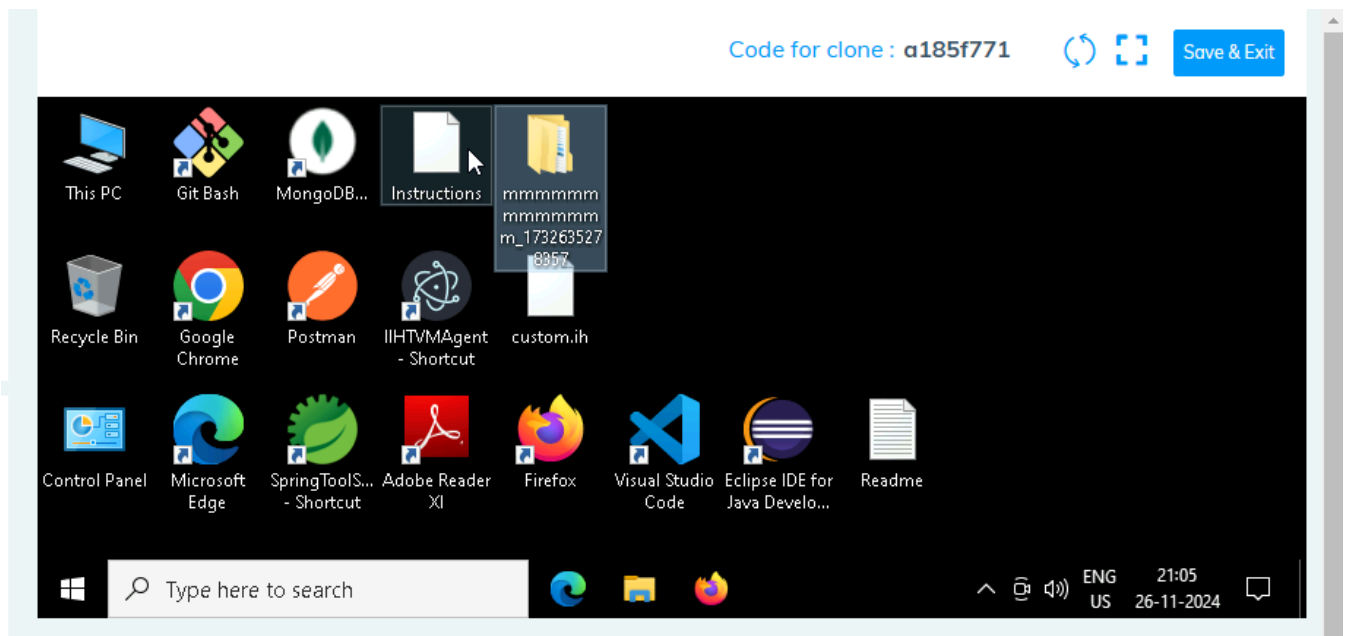
### 1.1 CODE QUALITY/OPTIMIZATIONS

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

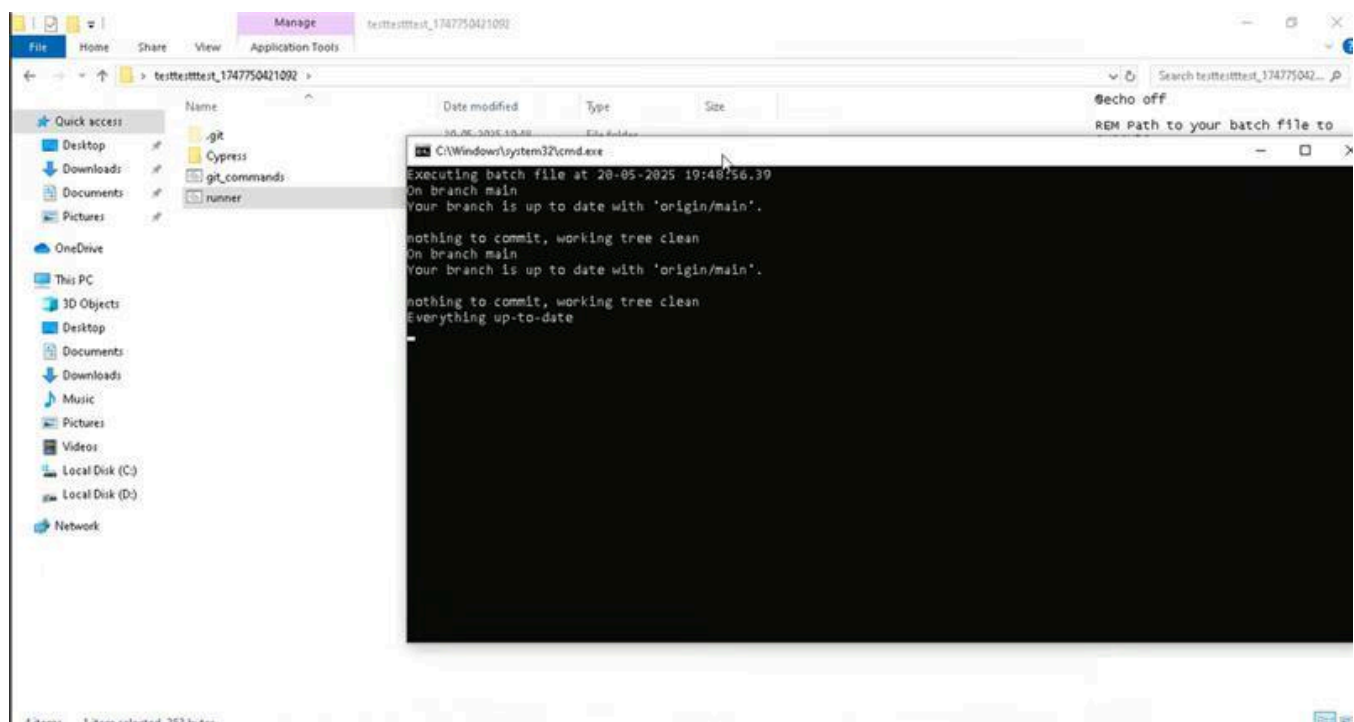
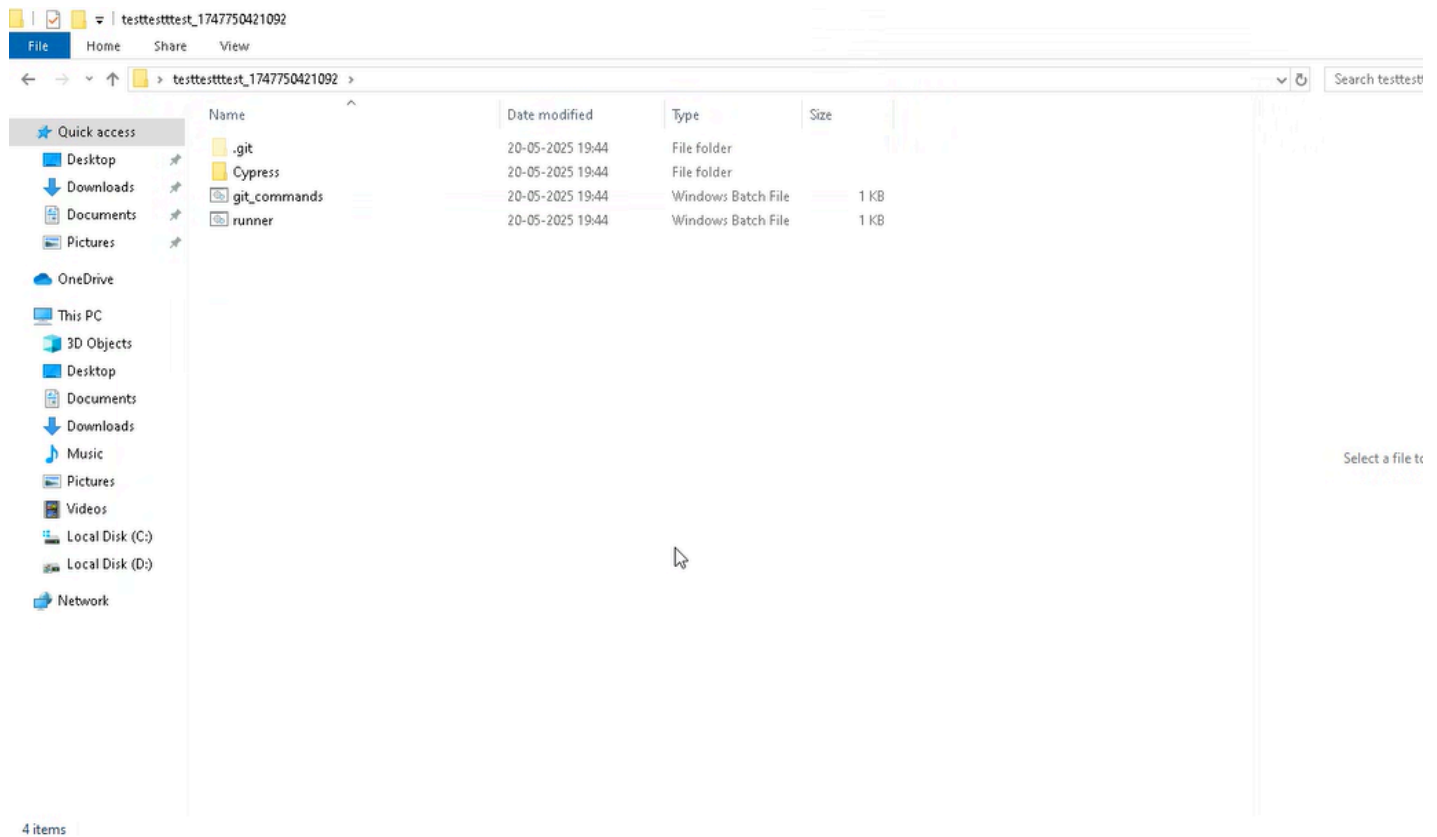
**Execution Steps:**

**Steps for Execution:**

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

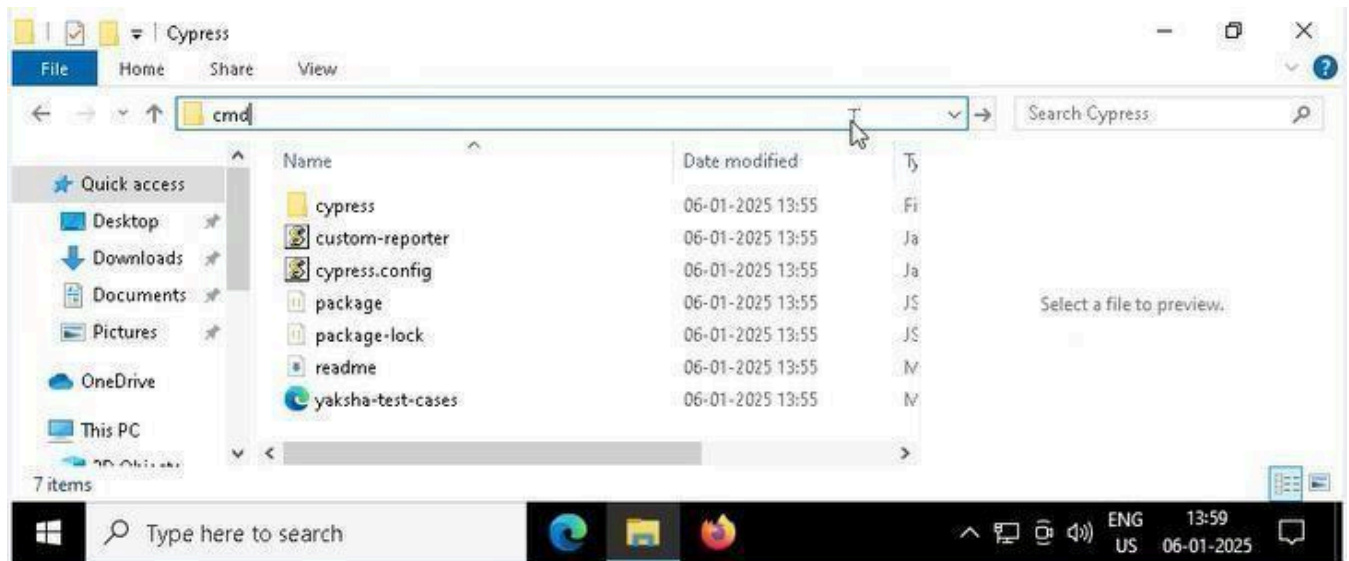


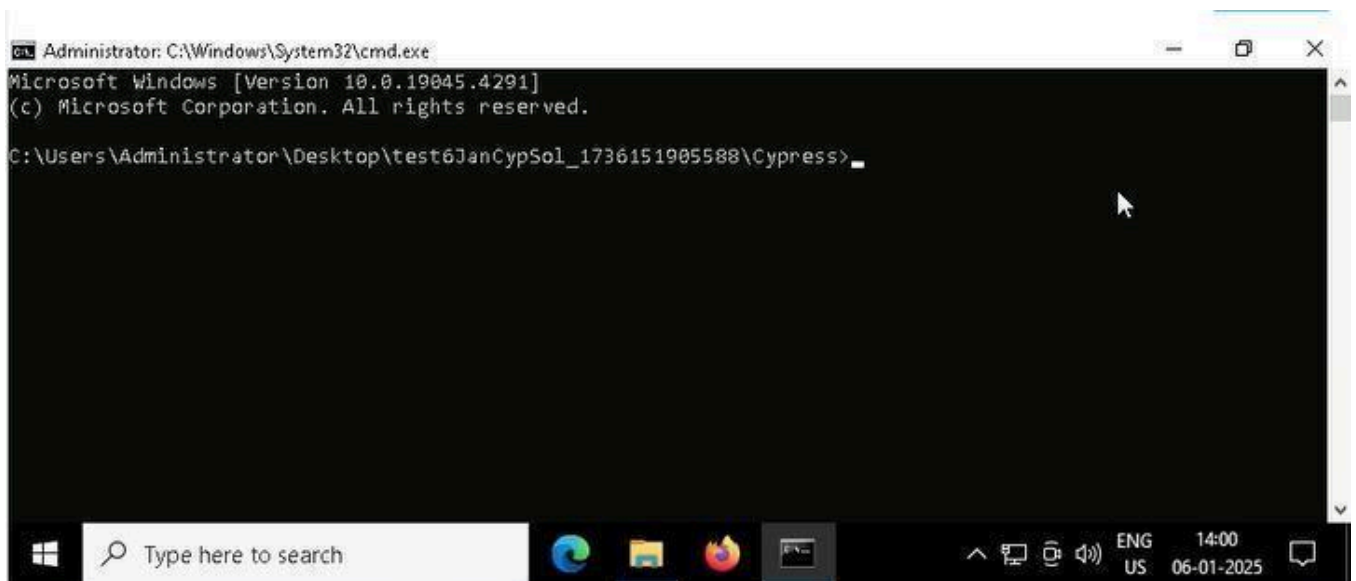
2. Execute this “runner” file. This will keep pushing the code at regular intervals.





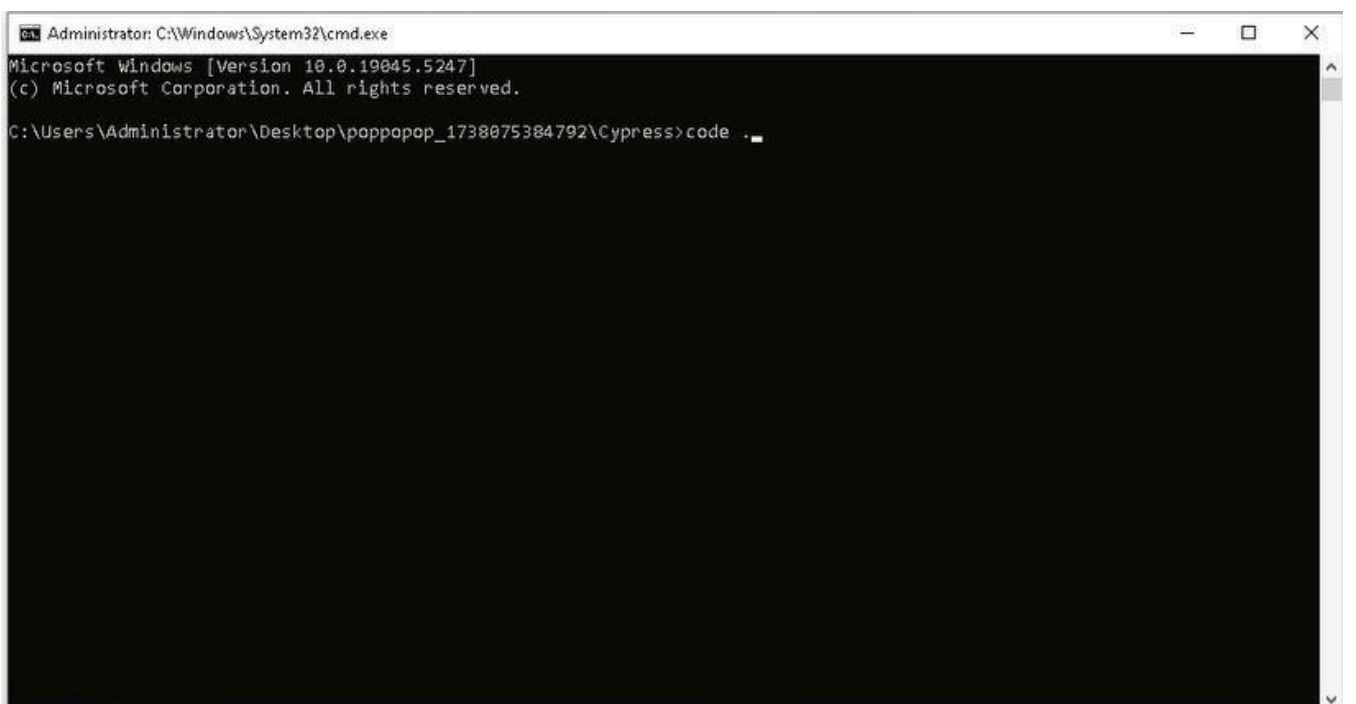
3. Go into the Cypress folder and Open command prompt with its location using “cmd” in search bar



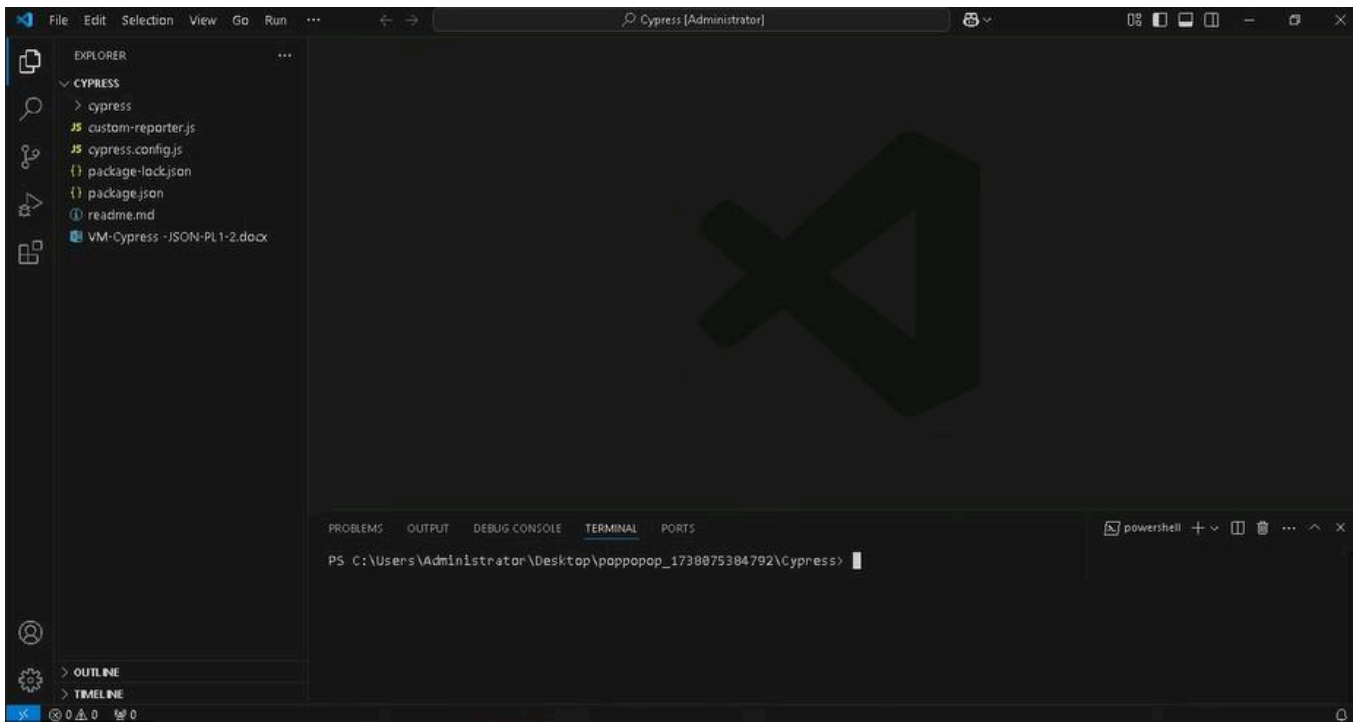


4. Open VS Code through cmd using below

command: code .



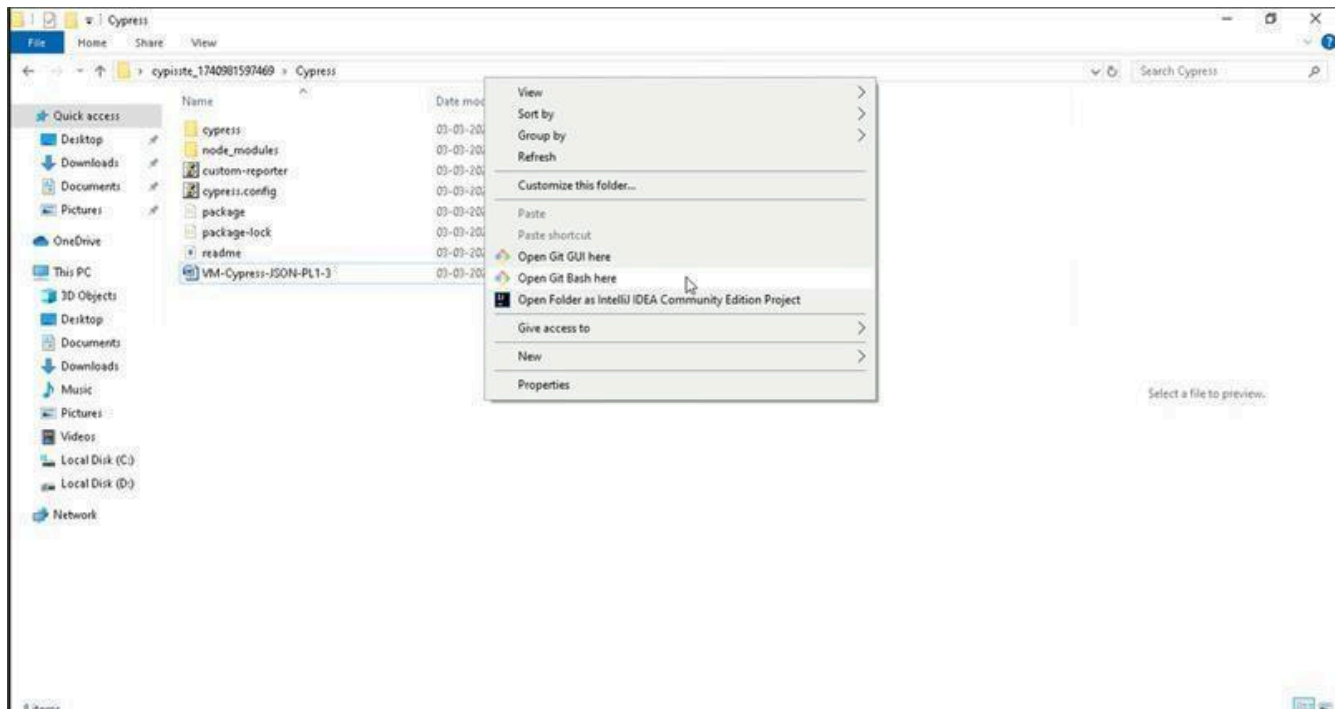
5. Once VsCode is open. Please open the terminal in Cypress folder:



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`

**\*\*NOTE: - It is a must to execute `npx cypress run`, this runs test cases in headless mode, shows summary in the end and submits the test cases on server.**

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:



and then execute commands like:

