YAKSHA

APP WITH JAVASCRIPT AND CYPRESS PL1-8

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 LoginData.json Etc.

Path	File	Description
\fixtures	LoginData.json TestCase_9.pdf	1. Contains data to read from json file.

PageObjects\Pages	● LoginProfilePage ● ProfilePage	1. All core activities 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. 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 Successfully	1. Login using valid credentials 2. 2. Click on "My Info" tab **Images must exist in cypress/fixtures** **User must be logged in.**	**Located in** /ProfilePage.js, spec.js **Methods:** myInfoTabAppear()	Verify the My Info tab loads successfully
2	Verify the 'Required' field error Message is displayed on leaving name field blank	1. Login using valid credentials 2. 2. Click on "My Info" tab 3. 3. Clear the name present in the name inputbar **User must be logged in with Admin privileges.**	**Reference path** e2e/ProfilePage.js: **Methods:** checkRequiredFieldError()	Required' field error message should be displayed
3	Verify the name gets edited successful ly	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 **User must be logged in with Admin access.**	**Reference path** e2e/ProfilePage.js: **Methods** editProfileName()	Verify the name gets updated in the top right corner

OrangeHRM automation using

4	Verify the 'My Info' tab's subtabs have unique URL	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')
		User must be logged in and on the home page
			Reference path
			e2e/ProfilePage.js
			Methods
			checkPersonalTab()
			checkContactTab()
			checkEmergencyTab()
				Verify the url from the three is unique

5	Verify new Dependant 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 rela **User must be logged in with Admin privileges.**	**Reference path** e2e/ProfilePage.js **Methods** assignDependentRecord()	Verify the presence of the child in the list
6	Verify new Dependant '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 **Images must exist in cypress/fixtures** **User must be logged in**	**Located in** e2e/ProfilePage.js: **Methods:** checkRelationOtherInputBar()	Verify the specify inputbar is not prsent before entering the other in relationship inputbar
7	Verify the dependants could be edited from the list	1. Login using valid credentials 2. Click on "My Info" tab 3. Navigate to 'Dependents' subtab 4. Create a new dependant using porevious steps 5. click the edit icon and change the name of the dependant **At least one user should exist**	**Reference path** e2e/ProfilePage.js: **Methods:** editDependentRecord();	Verify the dependant details get updated

		in the user list User must be logged in with Admin privileges.		
8	Verify the dependant's could be deleted from the list	1. Login using valid credentials 2. Click on "My Info" tab 3. Navigate to 'Dependents' subtab 4. Create a new dependant using porevious steps 5. click the delete icon and verify the candidate gets deleted from the list **User must be logged in with Admin access At least two records with different Usernames must exist**	**Reference path** e2e/ProfilePage.js: **Methods** deleteDependentRecord()	Verify the dependant details get deleted
9	Verify the sample pdf file could be uploaded in the attachments	1. Login using valid credentials 2. Click on "My Info" tab 3. Navigate to 'Dependents' subtab 4. Click on 'Add' Attachments button 5. Browse the pdf from the device 6. Click save button **User must be logged in and on the home page**	**Reference path** e2e/ProfilePage.js **Methods** uploadDependentPDF()	Verify the pdf gets uploaded
10	Verify the Comment inputbar has limit on length	1. Login using valid credentials 2. Click on "My Info" tab 3. Navigate to 'Dependents' subtab 4. Click on 'Add' Attachments button 5. Click Comments input bar and add really long text **User must be logged in and on the home page**	**Reference path** e2e/ProfilePage.js **Methods** checkForCommentLengthErrord()	verify the 'Should not exceed 200 characters' is displayed

OrangeHRM automation using

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

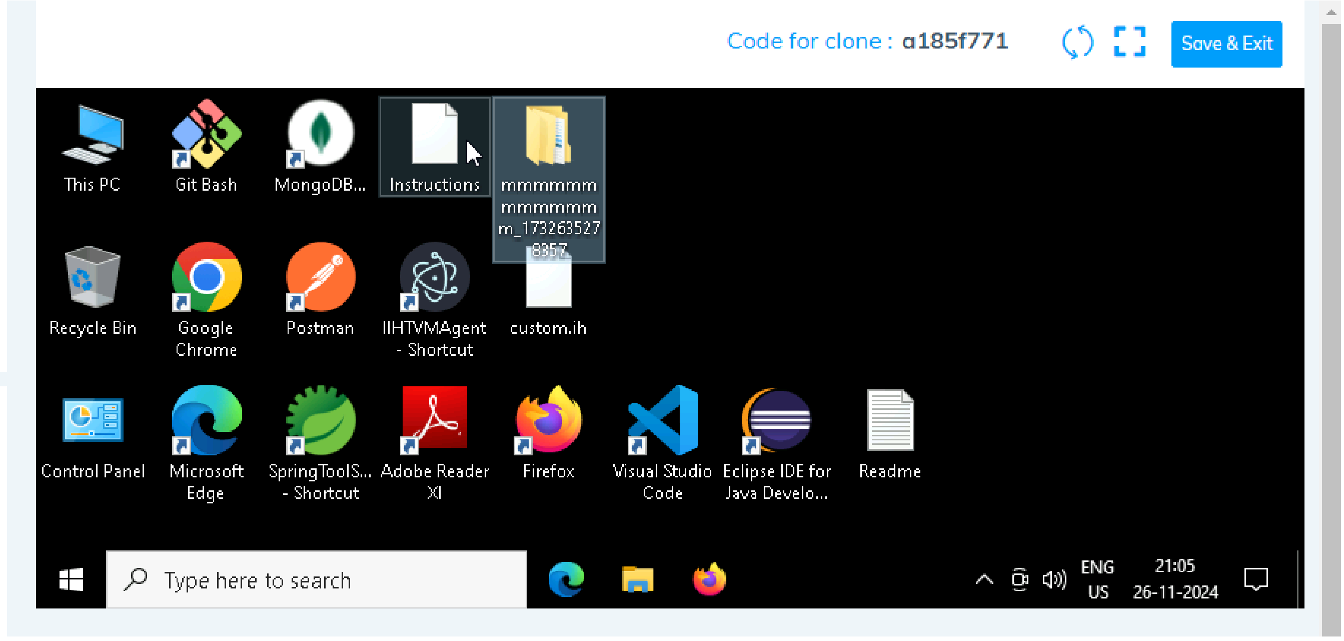
1.1 CODE QUALITY/OPTIMIZATIONS

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

Execution Steps:

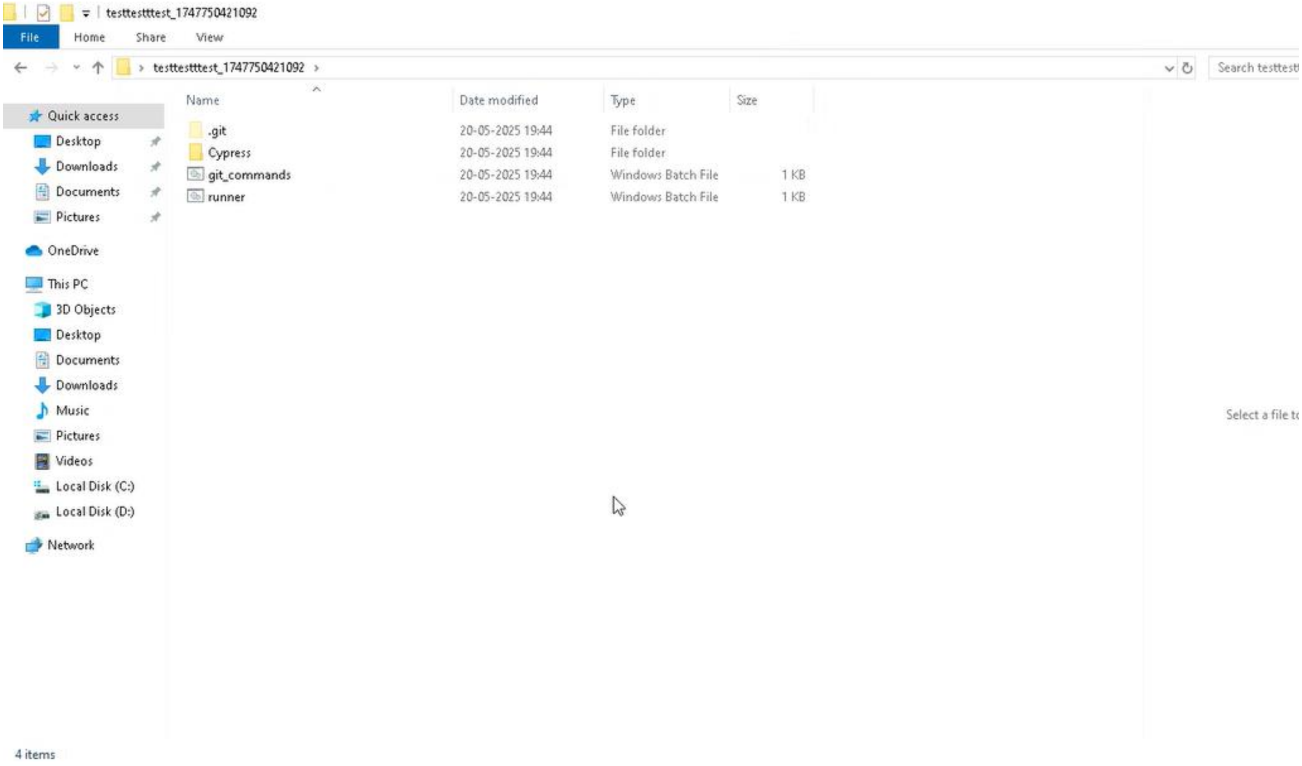
Steps for Execution:

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



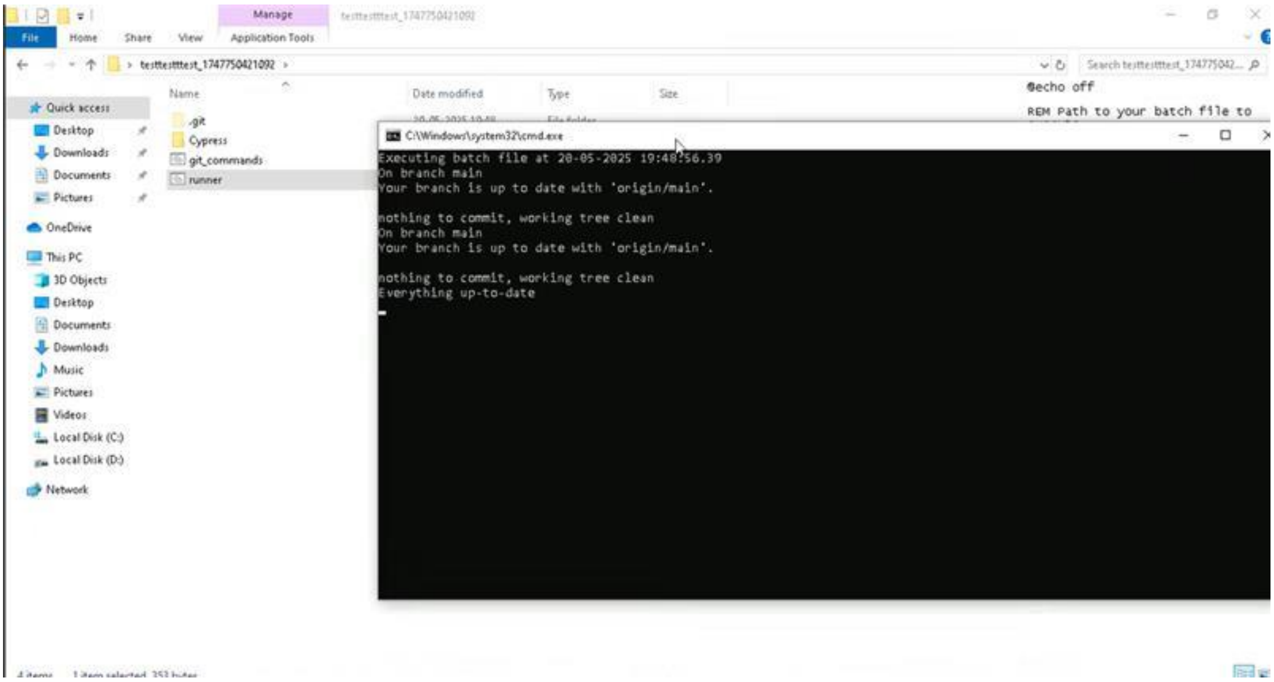
Code for clone : a185f771

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



testtestttest_1747750421092

Name	Date modified	Type	Size
.git	20-05-2025 19:44	File folder	
Cypress	20-05-2025 19:44	File folder	
git_commands	20-05-2025 19:44	Windows Batch File	1 KB
runner	20-05-2025 19:44	Windows Batch File	1 KB



```
@echo off

REM Path to your batch file to
```

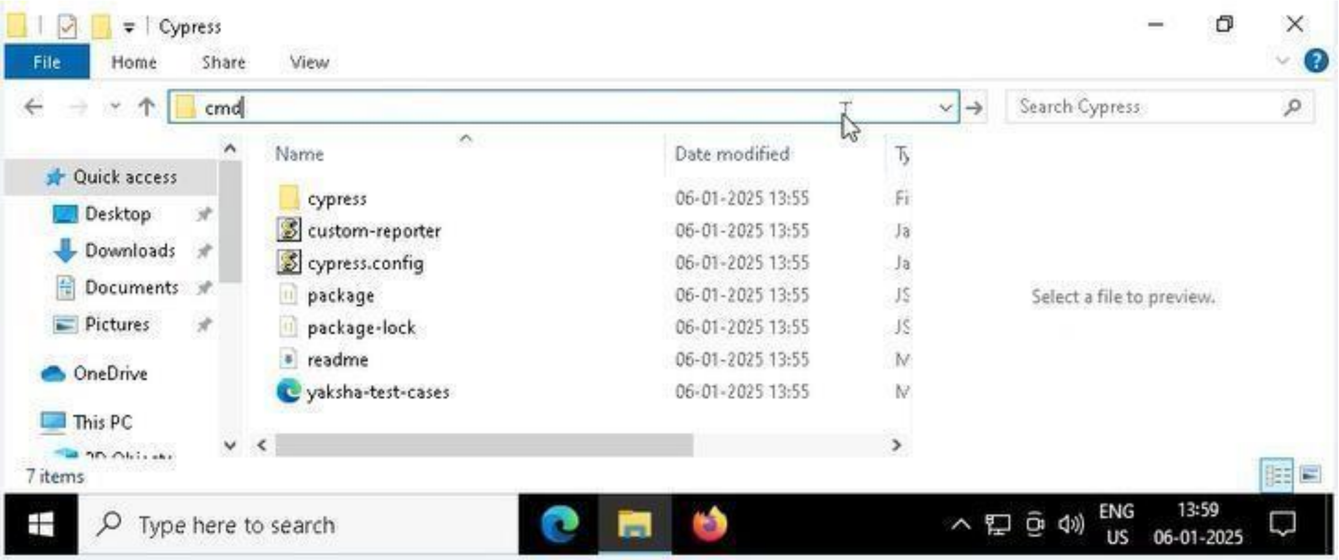
C:\Windows\system32\cmd.exe

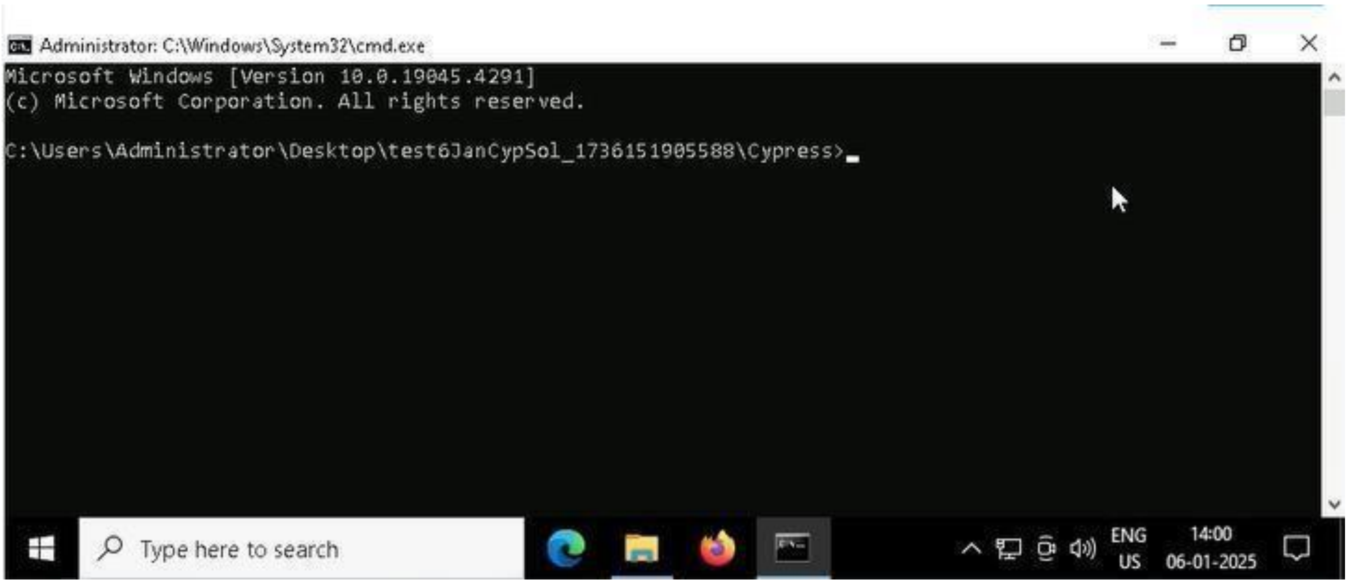
```
Executing batch file at 20-05-2025 19:48:56.39
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
Everything up-to-date
```

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



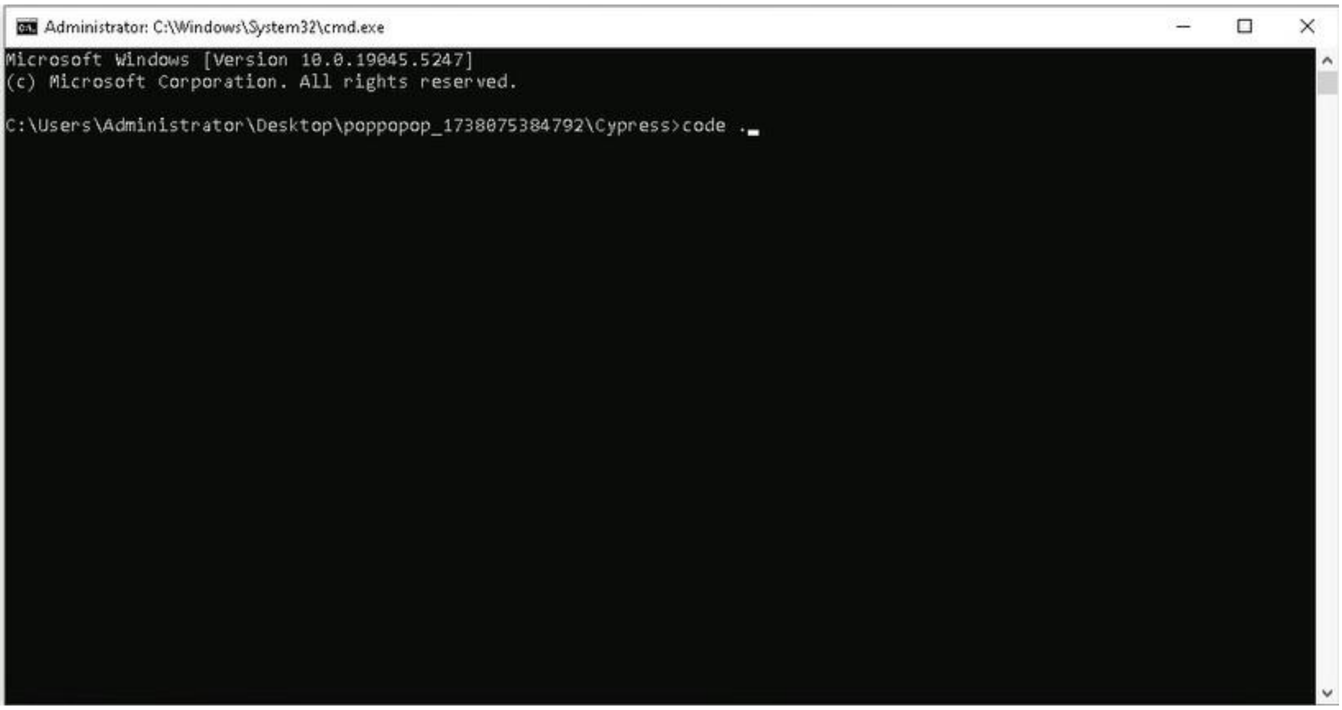


```
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Desktop\test6JanCypSol_1736151905588\Cypress>
```

4. **Open VS Code through cmd using below command:**

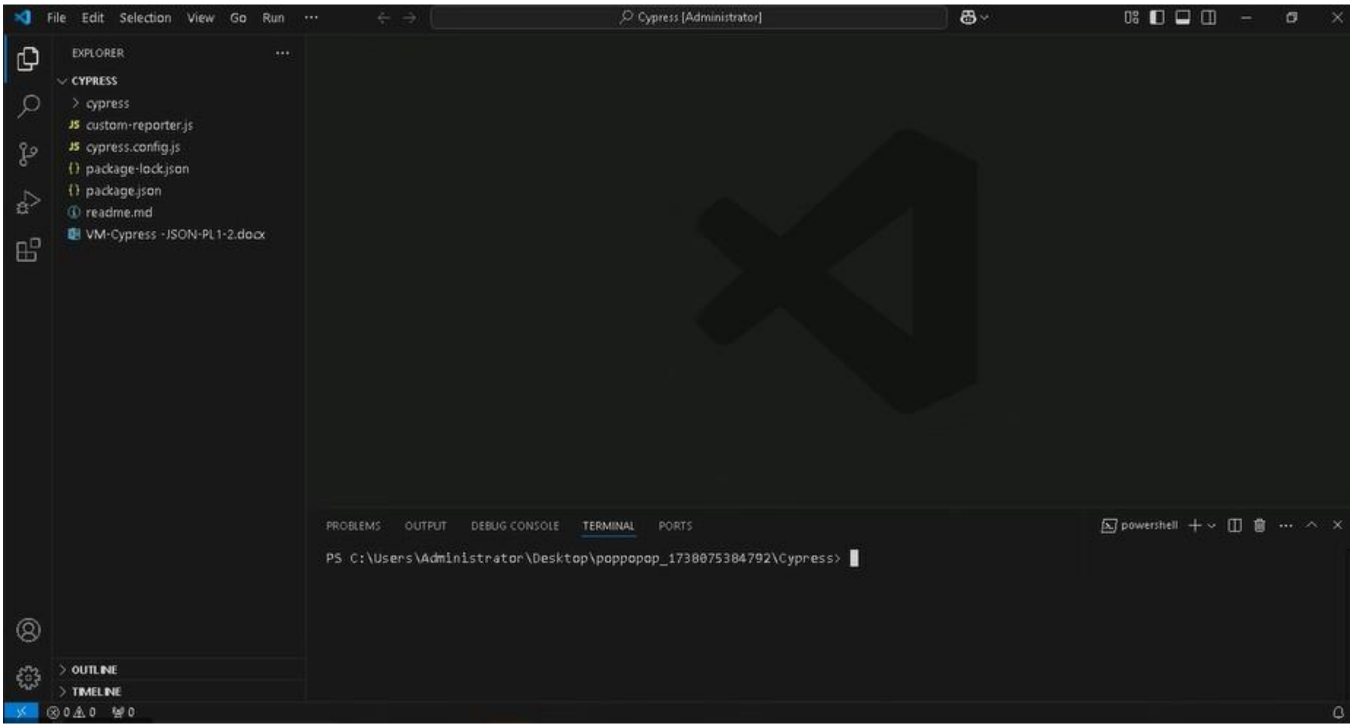
code .



```
Microsoft Windows [Version 10.0.19045.5247]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Desktop\poppopop_1738075384792\Cypress>code .
```

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



```
PS C:\Users\Administrator\Desktop\poppopop_1738075384792\Cypress>
```

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

   npm install

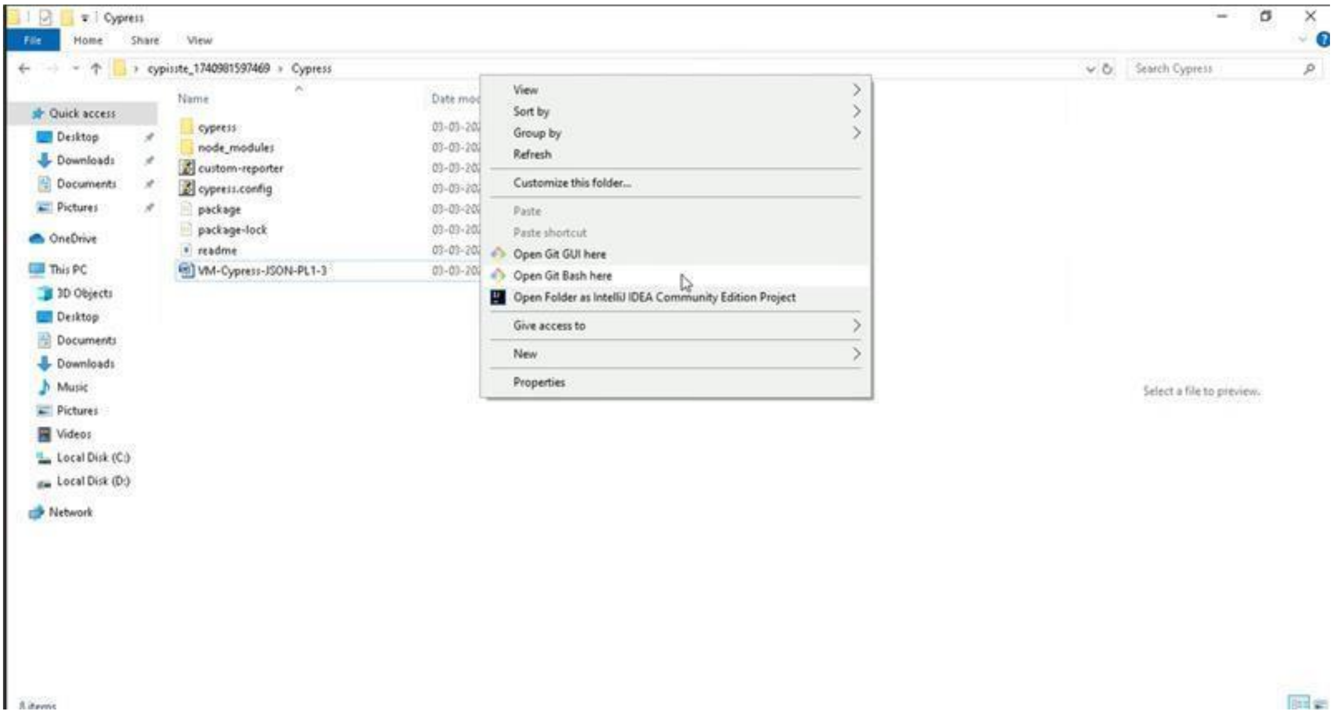
2. **Run the following command to open the interactive Cypress Test Runner in the Cypress folder path:**

   npx cypress open

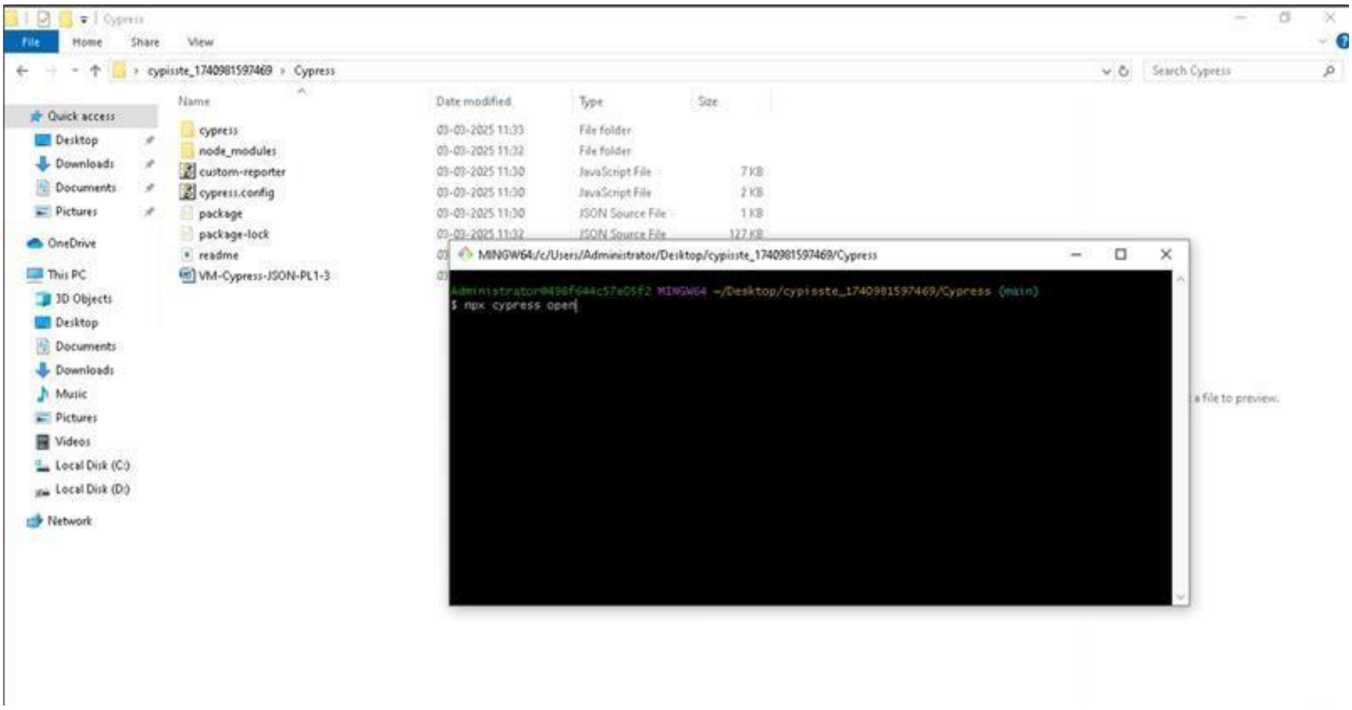
3. **Run the following command to run test cases in headless mode in the Cypress folder path:**

   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:



and then execute commands like:



Name	Date modified	Type	Size
cypress	03-03-2025 11:33	File folder	
node_modules	03-03-2025 11:32	File folder	
custom-reporter	03-03-2025 11:30	JavaScript File	7 KB
cypress.config	03-03-2025 11:30	JavaScript File	2 KB
package	03-03-2025 11:30	JSON Source File	1 KB
package-lock	03-03-2025 11:32	JSON Source File	127 KB
readme	03		
VM-Cypress-JSON-PL1-3	03		

MINGW64:/c/Users/Administrator/Desktop/cypisste_1740981597469/Cypress

```
Administrator@496f644c57e05f2 MINGW64 ~/Desktop/cypisste_1740981597469/Cypress (main)
$ npx cypress open
```