

# YAKSHA HEALTH APP WITH JAVASCRIPT AND CYPRESS

Mymedic automation using cypress

## 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	LoginData.json skylImage.jpg	1. Contains data to read from json file.

Mymedic automation using cypress

PageObjects\Pages	<ul style="list-style-type: none"> <li>• Admin</li> <li>• HomPage</li> <li>• LoginPage</li> <li>• ProfilePage</li> <li>• BuzzPage</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 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.</li> <li>4. Click Save</li> <li>5. Verify updated image using verifyImageReplaced()</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> <p>uploadProfilePicture('your-image.jpg') verifyProfilePictureUpdated()</p>	Profile image is successfully updated and visible
2	Verify Admin Can Edit Record	<ol style="list-style-type: none"> <li>1. Navigate to <b>Admin Page</b>.</li> <li>2. Click <b>Edit</b> button on the first listed user.</li> <li>3. Verify edit form appears.</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> <p>editUserFormAppear(); verifyUserEditFormAppears</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 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></p> <p>PageObjects/Pages/AdminPage.js:</p> <p><b>Methods</b></p> <p>goToAdminAndSortByUsername(); verifyUsernamesAreSortedAsc();</p>	The list of usernames should be sorted in ascending order after clicking the sort button
4	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></p> <p>PageObjects/Pages/HomePage.js</p> <p><b>Methods</b></p> <p>NavigateToAdminPage(); verifyUpgradeLinkAttributes();</p>	<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>
5	Verify on Hover Shows Help Text	<ol style="list-style-type: none"> <li>1. Navigate to the <b>Admin Page</b></li> <li>2. Locate the Help icon.</li> <li>3. Trigger hover</li> <li>4. Verify title="Help" or tooltip text appears</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>visitAdminForTollltip(); verifyHoverWork();</p>	The Help icon should display tooltip or attribute containing the text " <b>Help</b> ".

Test Case No.	Test Case Name	Test Steps to be performed	Path & Method Used	Expected Result
6	Verify Admin Can Search Users by Role	<ol style="list-style-type: none"> <li>1. Navigate to <b>Admin Page</b></li> <li>2. Open User Role dropdown</li> <li>3. Select <b>Admin</b></li> <li>4. Click Search</li> <li>5. Validate filtered results.</li> <li>6. <b>User must be logged in with valid credentials and have access to the Admin Page</b></li> </ol>	<b>Reference path</b>  PageObjects/Pages/AdminPage.js  <b>Methods</b>  searchAdminUsersByRole(); verifyAdminUsersDisplayed();	The user list should only display users with <b>Admin</b> role
7	Verify Image Upload and Share via Drag and Drop	<ol style="list-style-type: none"> <li>1. Go to <b>Buzz</b> tab</li> <li>2. Click <b>Share Photo</b> button</li> <li>3. Perform drag-and-drop of image</li> <li>4. Add text to post</li> <li>5. Click <b>Share</b></li> <li>6. Verify image is visible in shared post.</li> </ol> <b>User must be logged in and navigated to the Buzz page</b>	<b>Reference path</b>  PageObjects/Pages/BuzzPage.js  <b>Methods</b>  postImageWithTimestampText() ; verifyBuzzPostAppeared();	Image is successfully uploaded and displayed with post.
8	Verify Like Button is Functional	<ol style="list-style-type: none"> <li>1. Navigate to <b>Buzz</b> tab</li> <li>2. Wait for the first post to load</li> <li>3. Capture current like count</li> <li>4. Click the heart/like button</li> <li>5. Verify like count increased by 1.</li> </ol> <b>User must be logged in and at least one post must be visible in Buzz.</b>	<b>Reference path</b> PageObjects/Pages/BuzzPage.js:  <b>Methods :</b>  getFirstPost(); getLikeCount(); clickLikeButton(); verifyLikeCountReverse();	Like count should increment by 1 on clicking the heart Icon.
9	Verify Comment could be added to a post	<ol style="list-style-type: none"> <li>1. Navigate to <b>Buzz</b> tab</li> <li>2. Click comment icon on the first post</li> <li>3. Type a comment</li> <li>4. Press Enter or submit</li> <li>5. Verify success toast.</li> </ol> <b>User must be logged in and at least one post should be present on the Buzz page</b>	<b>Reference path</b>  PageObjects/Pages/BuzzPage.js;  <b>Methods:</b>  addCommentToFirstPost();  verifyCommentSuccess();	The comment should be added successfully, and a "Successfully Saved" toast message should appear

Test Case No.	Test Case Name	Test Steps to be performed	Path & Method Used	Expected Result
10	Verify Buzz Post Editing Function-ality	<ol style="list-style-type: none"> <li>1. Navigate to <b>Buzz</b> tab</li> <li>2. Click the <b>three-dot menu</b> on the first post</li> <li>3. Select <b>Edit Post</b></li> <li>4. Modify the post content</li> <li>5. Click <b>Post</b> to save changes</li> <li>6. Verify <b>success toast</b> appears.</li> </ol> <p><b>Note :- User must be logged in and must have permission to edit their own Buzz post</b></p>	<p><b>Reference path</b></p> <p>PageObjects/Pages/BuzzPage.js</p> <p><b>Methods</b></p> <pre>editFirstPostWithText(); verifyPostEditSuccess();</pre>	<p>A success confirmation popup with the message: <b>"Imaging and lab order add successfully"</b> should appear.</p>

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://healthapp.yaksha.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

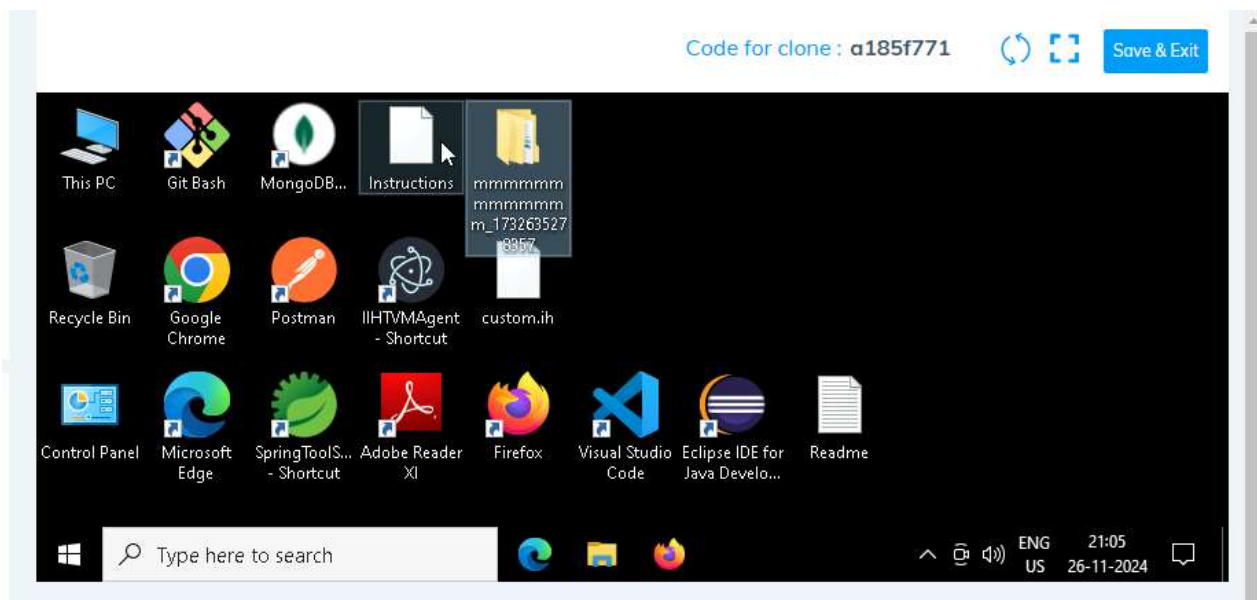
### 1.1 CODE QUALITY/OPTIMIZATIONS

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

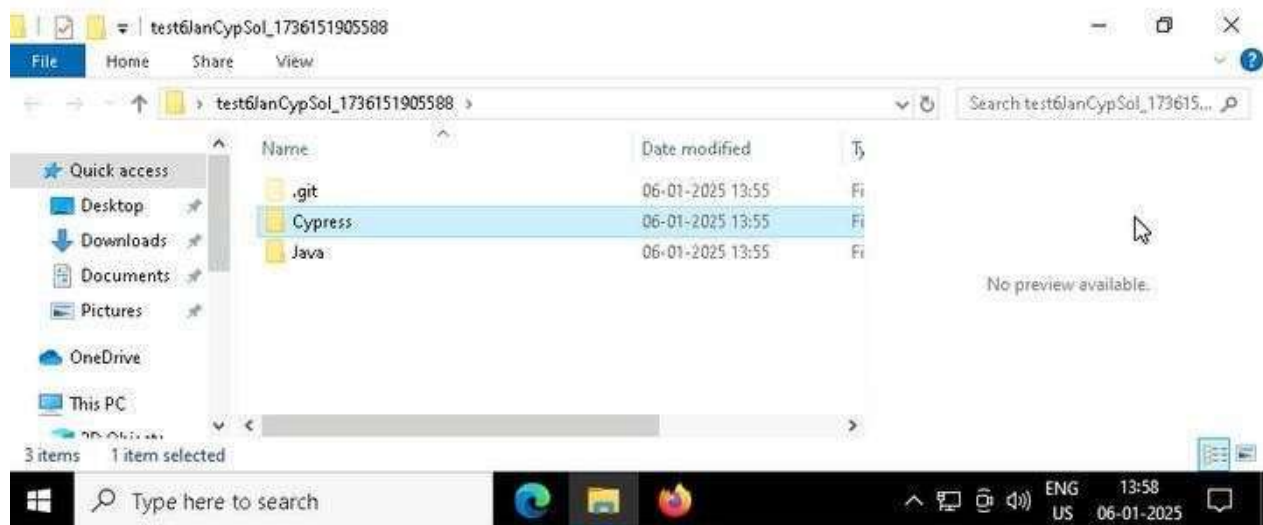
#### Execution Steps:

##### Steps for Execution:

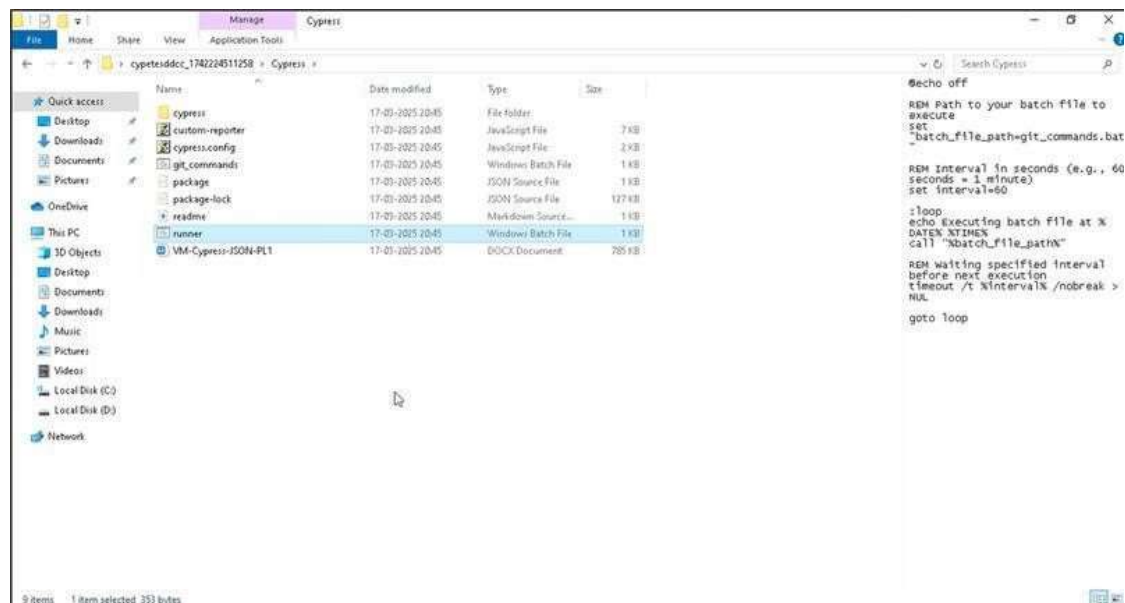
1. Please open the folder created on desktop with the email name you used to login.



Mymedic automation using cypress

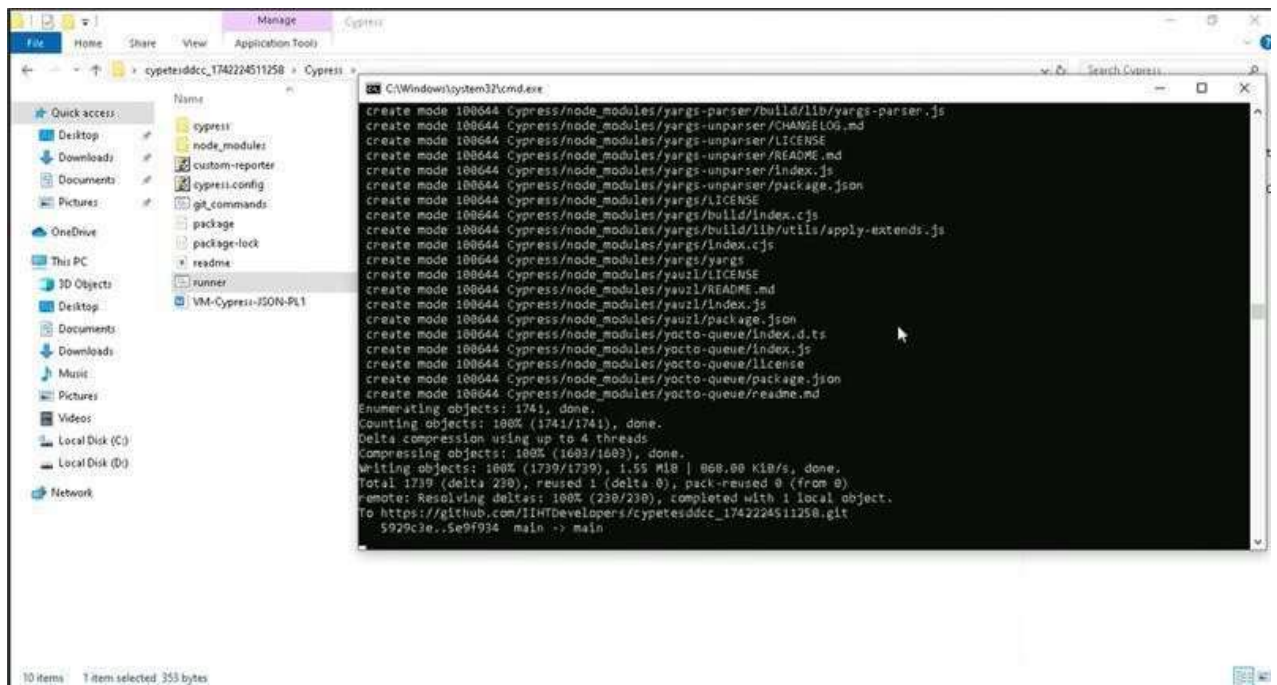


- Go into the Cypress folder and execute this “runner” file. This will keep pushing the code at regular intervals.

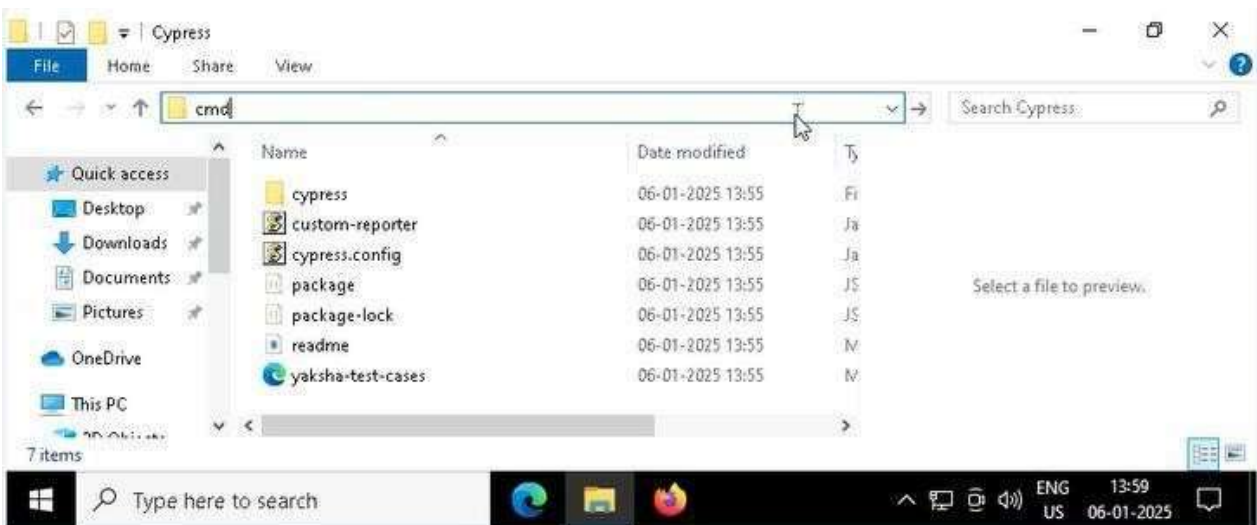


Mymedic automation using cypress

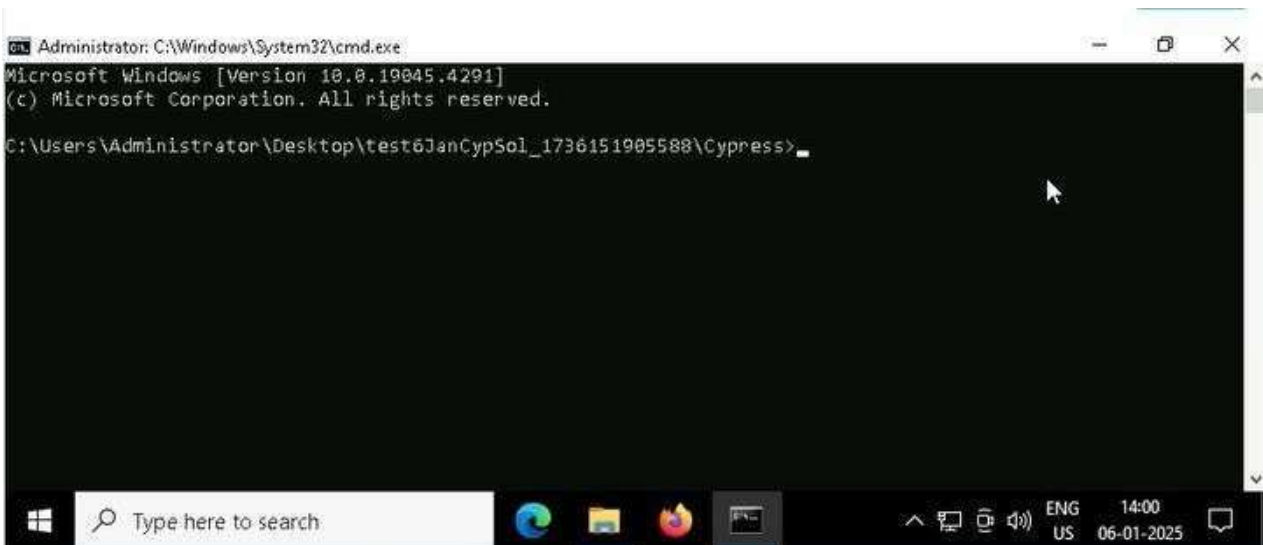




### 3. Open command prompt with its location using "cmd" in search bar



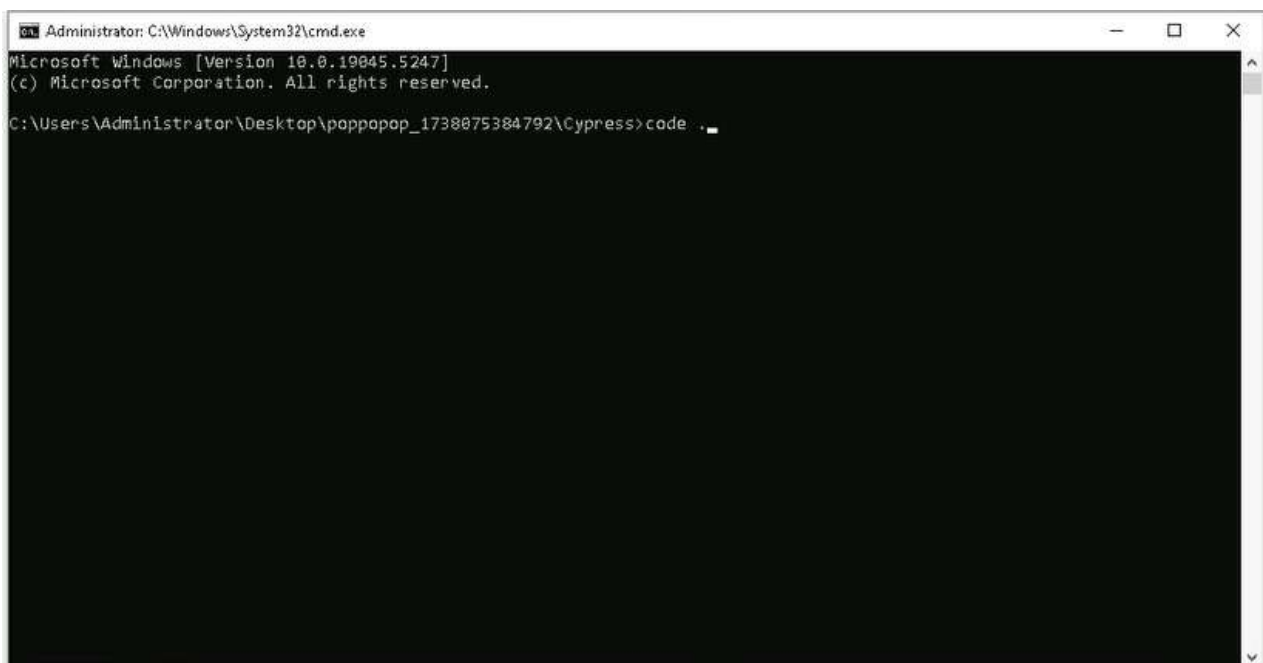
Mymedic automation using cypress



A screenshot of a Windows command prompt window titled "Administrator: C:\Windows\System32\cmd.exe". The window shows the following text: "Microsoft Windows [Version 10.0.19045.4291] (c) Microsoft Corporation. All rights reserved. C:\Users\Administrator\Desktop\test6JanCypSol\_1736151905588\Cypress>". The taskbar at the bottom shows the Windows logo, a search bar, and several application icons. The system tray on the right indicates the language is "ENG US", the time is "14:00", and the date is "06-01-2025".

4. Open VS Code through cmd using below command:

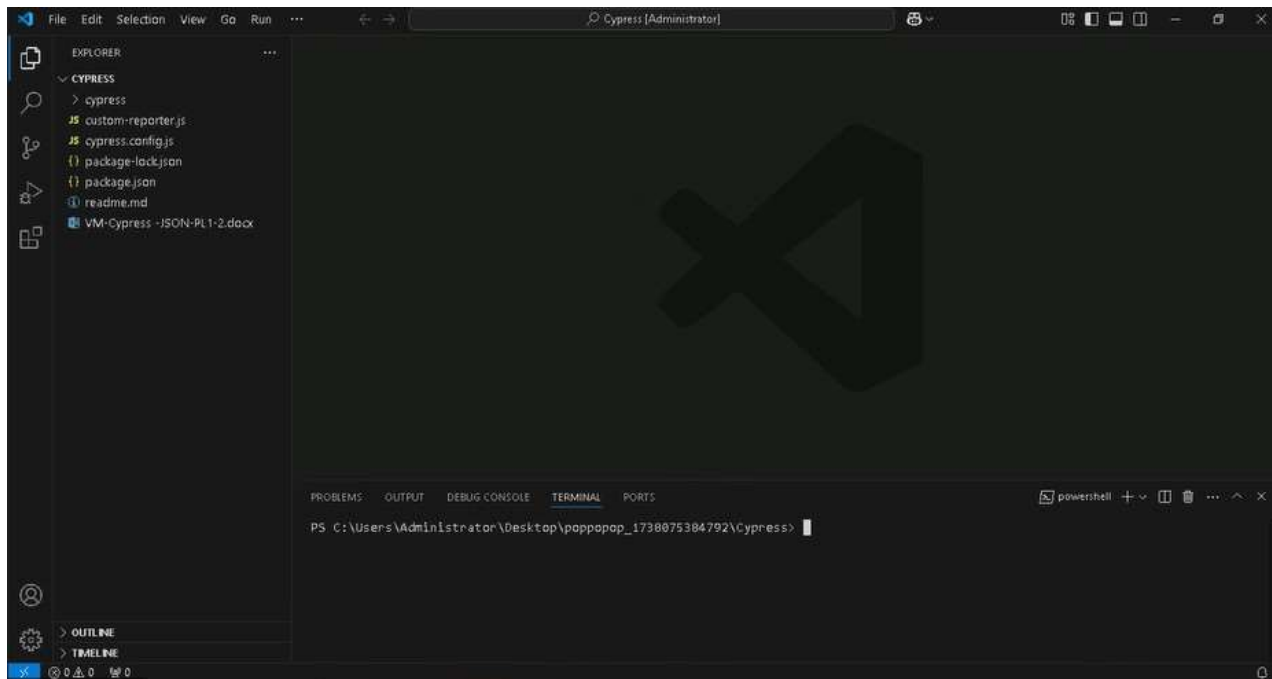
code .



A screenshot of a Windows command prompt window titled "Administrator: C:\Windows\System32\cmd.exe". The window shows the following text: "Microsoft Windows [Version 10.0.19045.5247] (c) Microsoft Corporation. All rights reserved. C:\Users\Administrator\Desktop\poppopop\_1738075384792\Cypress>code .". The taskbar and system tray are visible at the bottom, showing the same date and time as the previous screenshot.

Mymedic automation using cypress

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



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

**npm install**

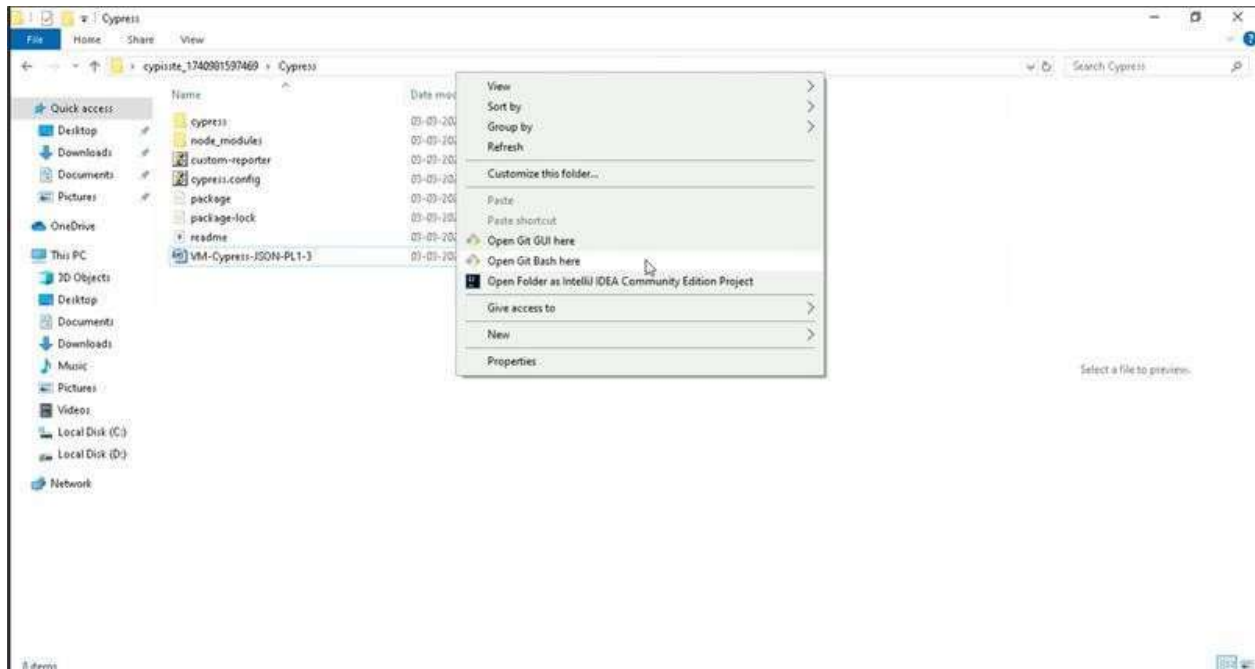
**2. Run the following command to open the interactive Cypress Test Runner in the Cypress folder path:**

**npx cypress open**

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

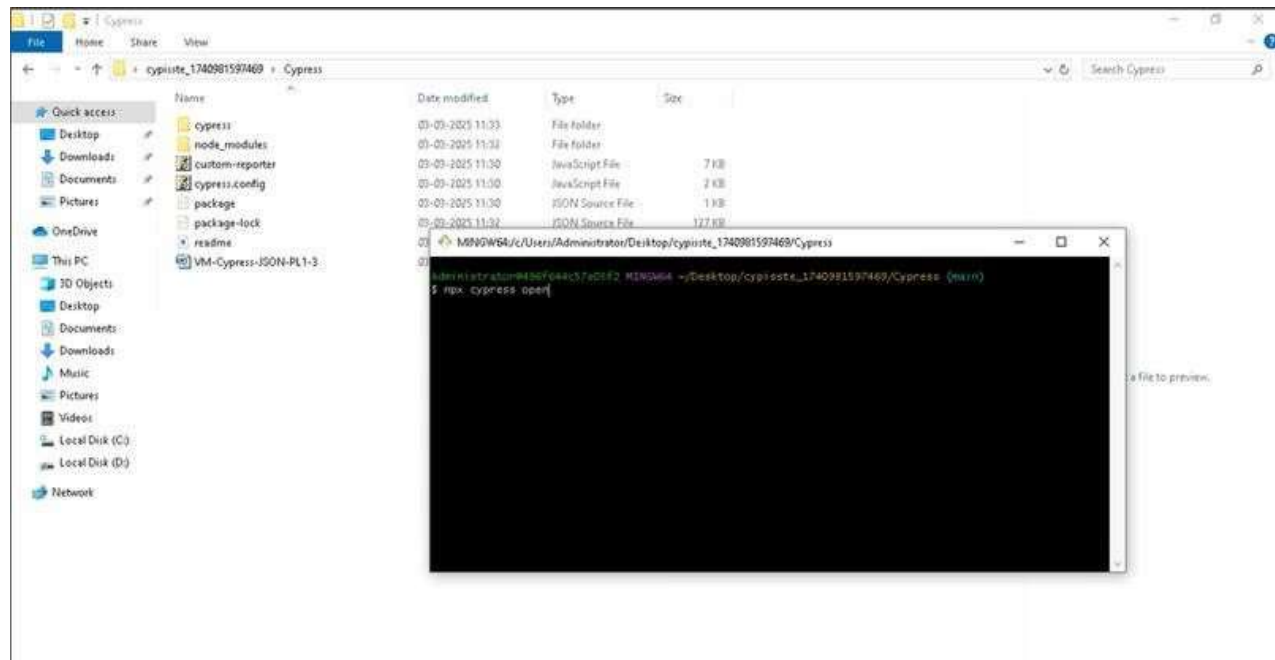
**npx cypress run**

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:



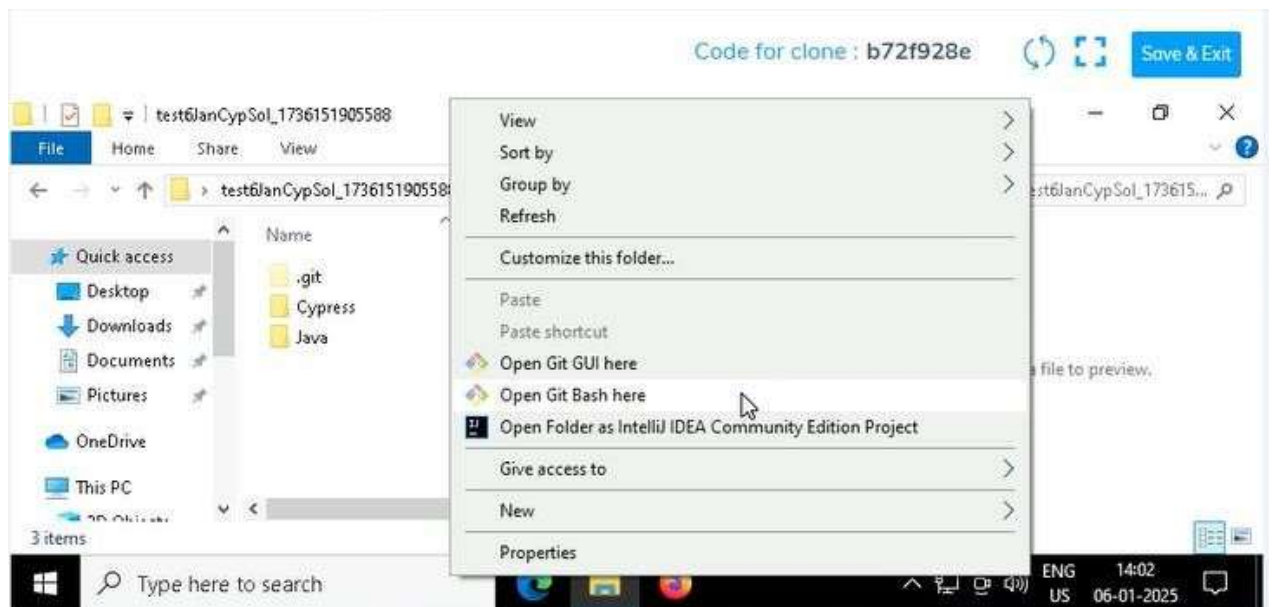
Mymedic automation using cypress

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



Mymedic automation using cypress

## 2. Add all files

Project

Speed Test Avg: 4.36Mbps Live: >= 10Mbps Time Remaining: 02:54:11 Final Submission

Question

Q1. VM-Playwright-Excel-Solution

Instructions

Introduction

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.

Launch

an launch VM by clicking "Launch Assessment Environment". It will take around 5-10 mins to launch the

Code for clone : be5b9203

Save & Exit

MINGW64/c/Users/Administrator/Desktop/zxczxc\_1732782665185

```
Administrator@4f6d8dad9dbf5e2 MINGW64 ~/Desktop/zxczxc_1732782665185 (main)
$ git add .
```

## 3. Commit the changes

Code for clone : be5b9203

Save & Exit

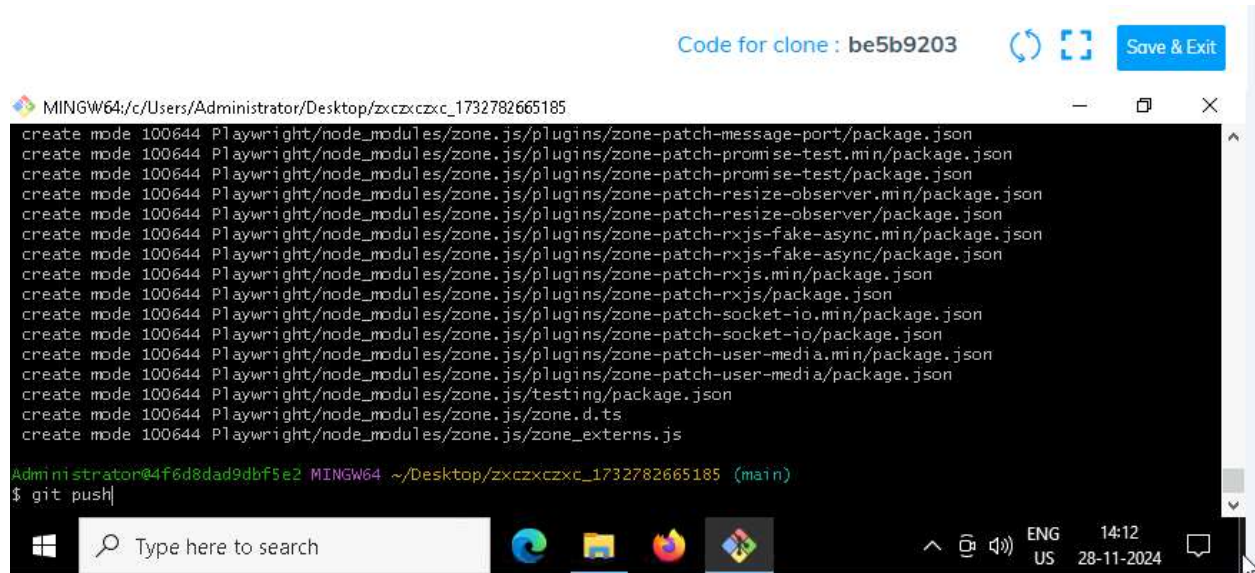
MINGW64/c/Users/Administrator/Desktop/zxczxc\_1732782665185

```
warning: in the working copy of 'Playwright/node_modules/zone.js/plugins/zone-patch-rxjs/package.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/zone.js/plugins/zone-patch-socket-io.min/package.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/zone.js/plugins/zone-patch-socket-io/package.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/zone.js/plugins/zone-patch-user-media.min/package.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/zone.js/plugins/zone-patch-user-media/package.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/zone.js/testing/package.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/zone.js/zone.d.ts', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Playwright/node_modules/zone.js/zone_externs.js', LF will be replaced by CRLF the next time Git touches it

Administrator@4f6d8dad9dbf5e2 MINGW64 ~/Desktop/zxczxc_1732782665185 (main)
$ git commit -m "committing the updates"
```

Mymedic automation using cypress

#### 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 directory. The files listed include zone-patch-message-port/package.json, zone-patch-promise-test.min/package.json, zone-patch-resize-observer.min/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-socket-io.min/package.json, zone-patch-user-media.min/package.json, zone-patch-user-media/package.json, zone-testing/package.json, zone.d.ts, and zone\_extrns.js. The terminal prompt shows the user is Administrator@4f6d8dad9dbf5e2 in the MINGW64 shell, located at ~/Desktop/zxczxcxc\_1732782665185 (main). The command \$ git push is entered. 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_extrns.js
Administrator@4f6d8dad9dbf5e2 MINGW64 ~/Desktop/zxczxcxc_1732782665185 (main)
$ git push
```

Mymedic automation using cypress