YAKSHA HEALTH APP WITH JAVASCRIPT AND CYPRESS

Mymedic automation using cypress

Usecase summary

Project Name: healthapp.yaksha app – Medical Record Management System

Use Case Summary: healthapp.yaksha is a healthcare application designed to manage Electronic Medical Records (EMR). it allows users to view, search, and manage patient records. It features functionality such as adding/editing patient records, filtering data by doctor and department, and exporting records. The primary use case is to automate the process of medical record management, ensuring efficient and reliable operations for healthcare providers.

Technology Stack:
- **Automation Tool:** Cypress (for testing)

Key Features:
- **Patient Record Management:** Add, edit, and delete patient records.
- **Filtering and Search:** Search medical records by date range, doctor, department, and more.
- **Export Functionality:** Export records for offline access.

Expected Outcomes:
- Automate key healthcare operations like patient record handling, filtering, and validation.
- Ensure the accurate retrieval and modification of medical records, enhancing operational efficiency.

Overview of the application

Pages/Features that are to be focused for the application

Please use the Application URL https://healthapp.yaksha.com

PROBLEM STATEMENT

Need to automate the following activities using cypress+javascript

You will be given few Json files in fixtures folder like dateRange.json, invalidJson.json, loginData.json and patientNames.json.

Path	File	Description
\fixtures	dateRange.json invalidJson.json loginData.json patientNames.json	1. Contains data to read from json files.
PageObjects\Pages	● ADTPage ● AppointmentPage ● DispensaryPage ● LaboratoryPage	1. All core activities to be performed here. 2. The comments associated with each templated method here

Mymedic automation using cypress

		● LoginPage		describe the expectation.
		● PatientPage		3. Declare any variable/object you need to share data/status between different methods.
		● ProcurementPage		
		● RadiologyPage		
		● SettingsPage		4. Do not modify the signature of methods declared here.
		● UtilitiesPage		

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
	Verify Login with Valid Credentials	1. The application should read the data from loginData.json file and fetch the user's name and password. 2. It should call the method performLogin() 3. Perform login method will perform authentication with the username and password. 4. Verify admin name is visible on the home page. **It is must to implement this login functionality at first and then implement any other test case.**	**Reference path** \PageObjects\Pages **LoginPage** **methods** performLogin()	Successfully logs in with provided credentials. The user is logged in the admin page.
1	Verify Page Navigation and Load Time for Billing Counter	1. Use verifyBillingCounterLoadState() to check module load. 2. Open "Change Billing Counter" module using ChangeBillingCounter. 3. Set acceptable load time: 1000ms. 4. Verify counter presence; select the first counter if available. 5. Log a message if no counters are found.	**Reference path** \PageObjects\Pages \ **UtilitiesPage** **methods** verifyBillingCounterLoadState()	Navigates to "Change Billing Counter". Proceeds if counters are available and loaded within 1 second; logs a message otherwise.
2	Patient Search with Valid Data	1. Navigate to Appointment page. 2. Use verifypatientName() to verify the first patient is present or not. 3. Patient name must be fetched from patientNames.json file. 4. Validate that search results contain the searched name.	**Reference path** \PageObjects\Pages \ **AppointmentPage** **methods** searchAndVerifypatientName()	Displays patient name in search results. Ensures short name matches the searched data.
3	Activate Counter in Dispensary	1. Use verifyActiveCounterMessageInDispensary() to: - Navigate to the Dispensary page. - Select a random counter if available. - Activate the counter and verify the activation message. - Navigate back to Counter tab. 2. Log counter selection and activation status.	**Reference path** \PageObjects\Pages \ **DispensaryPage** **methods** verifyActiveCounterMessageInDispensary()	Counter activation message matches the selected counter name.

Mymedic automation using cypress

Test Case No.	Test Case Name	Test Steps to be performed	Path & Method Used	Expected Result
4	Purchase Request List Load	1. Use verifyAllElementsVisible() to check verify of elements: purchaseRequest, purchaseOrder,goodsArrivalNotification, quotations, settings, reports, favoriteButton, okButton, printButton, firstButton, previousButton, nextButton, lastButton.	**Reference path** \PageObjects\Pages \ **ProcurementPage** **methods** verifyAllElementsVisible()	All elements are present and visible on the procurement page.
5	Verify Error Message While Adding New Lab Test	1. Navigate to Laboratory > Settings. 2. Select "Add New Lab Test". 3. Click "Add" without providing inputs. 4. Capture and verify the error message: "Lab Test Code Required".	**Reference path** \PageObjects\Pages \ **LaboratoryPage** **methods** verifyErrorMessage()	Error message: "Lab Test Code Required" is displayed.
6	Handle Alert on Radiology Module	1. Navigate to Radiology module and select "List Request" sub-module. 2. Apply filter using fromDate. 3. The application should read the data from dateRange.json file for fromDate and toDate. 3. Click "Add Report" button. 4. A pop-up will open. Then click on X button to close the pop-up. 5. Now an alert will show up. You need to handle that. 6. Verify and accept the alert if the message is "**Changes will be discarded. Do you want to close anyway?**". 7. The application should check for matched data.	**Reference path** \PageObjects\Pages \ **RadiologyPage** **methods** performRadiologyRequestAndHandleAlert()	Alert dialog matches expected message and is accepted. Return a Cypress chainable that resolves to true if the alert is handled successfully
7	Verify Patient Search Functionality with Multiple Patients	1. Navigate to Patient Section. 2. Fetch all patient names as an array from patientNames.json file. 3. Use the search bar to find patients. 4. Compare retrieved names with expected names from json file. 5. Log success or failure for each match.	**Reference path** \PageObjects\Pages**PatientPage** **methods** searchAndVerifyPatients()	Matches all patient names from Json.
8	Error Handling and Logging in Purchase Request List	1. Navigate to Procurement module. 2. Apply invalid date filter in format as **00-00-0000**. 3. Click "OK". 4. Capture and verify the error message: "Date is not between range. Please enter again."	**Reference path** \PageObjects\Pages \ **ProcurementPage** **methods** verifyNoticeMessageAfterEnteringIncorrectFilters()	Triggers and verifies the error message for invalid date range. Logs success for match; failure for mismatch.
9	Modular Script for Patient Search	1. Navigate to the Appointment section. 2. Use searchPatientInAppointment() to execute the search. 3. Validate the patient search result.	**Reference path**	You should be able to search patient search data.

Mymedic automation using cypress

Test Case No.	Test Case Name	Test Steps to be performed	Path & Method Used	Expected Result
		4. Navigate to the Patient section. 5. Use searchPatientInPatientPage() to execute the search. 6. Validate the patient search result. 7. Navigate to the ADT section. 8. Use searchPatientInADT() to execute the search. 9. Validate the patient search result.	\PageObjects\Pages \ **AppointmentPage** \PageObjects\Pages \ **PatientPage** \PageObjects\Pages \ **ADTPage** **methods** searchPatientInAppointment() searchPatientInPatientPage() searchPatientInADT() **We have created searchPatient() method present in reusableMethod file. You can use that also.**	
10	Verify 'Morning Counter' selection and report generation for the specified date	1. Navigate to dispensary module. 2. Click on Reports. 3. Click on User Collection report to navigate to the respective report section. 4. Read the fromDate from dtaeRange.json file and enter in From Date input field. 5. Select morning counter from dropdrown. 6. Click on Show Report button to generate the report. 7. Verify the generated report correctly displays the morning counter in counter column.	**Reference path** \PageObjects\Pages \ **DispensaryPage** **methods** generateMorningCounterReport()	Verify the generated report correctly displays the morning counter in counter column.
11	Verify the tooltip text on hover of Star icon in Laboratory	1. Navigate to Laboratory module. 2. Hovers over star icon and waits for tooltip to appear. 3. Verifies the visibility of the star icon and retrieves the tooltip text.	**Reference path** \PageObjects\Pages \ **LaboratoryPage** **methods** verifyStarTooltip()	Verifies the visibility of the star icon and retrieves the tooltip text. Result should be "Remember this date".
12	Add and Verify New Imaging Type in Radiology	1. Navigates to the Settings module and clicks on the Radiology submodule. 2. Clicks on the "Add Imaging Type" button to open the modal for adding a new imaging type. 3. Fills the "Imaging Item Name" field with a random name (Test-{random4digitnumber}) and clicks "Add". 4. Verifies that the newly added imaging type appears in the list of imaging types.	**Reference path** \PageObjects\Pages \ **SettingsPage** **methods** addAndVerifyNewImagingType()	Verifies that the newly added imaging type appears in the list of imaging types.
13	Web Element Handling for Dropdowns in Purchase Request	1. Navigate to Purchase Request List. 2. Fetch the fromDate and toDate from dateRange.json file. 3. Apply date range filter. 4. Retrieve dates from "Requested Date" column and validate each date within range. 5. Wait for success or failure.	**Reference path** \PageObjects\Pages \ **ProcurementPage** **methods** verifyRequestedDateColumnDateWithinRange()	It should be able to retrieve dates are within range.

Mymedic automation using cypress

Test Case No.	Test Case Name	Test Steps to be performed	Path & Method Used	Expected Result
14	Verify logout functionality from Admin dropdown	1. Navigate to index page i.e https://healthapp.yaksha.com/Home/Index#/ 2. Click on the "Admin" dropdown 3. Click on "Log Out" option. 4. Verify the user is redirected to the login page.	**Reference path** \PageObjects\Pages\LoginPage **methods** verifyLogoutFunctionality()	User is logged out successfully and the login page is displayed.

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.

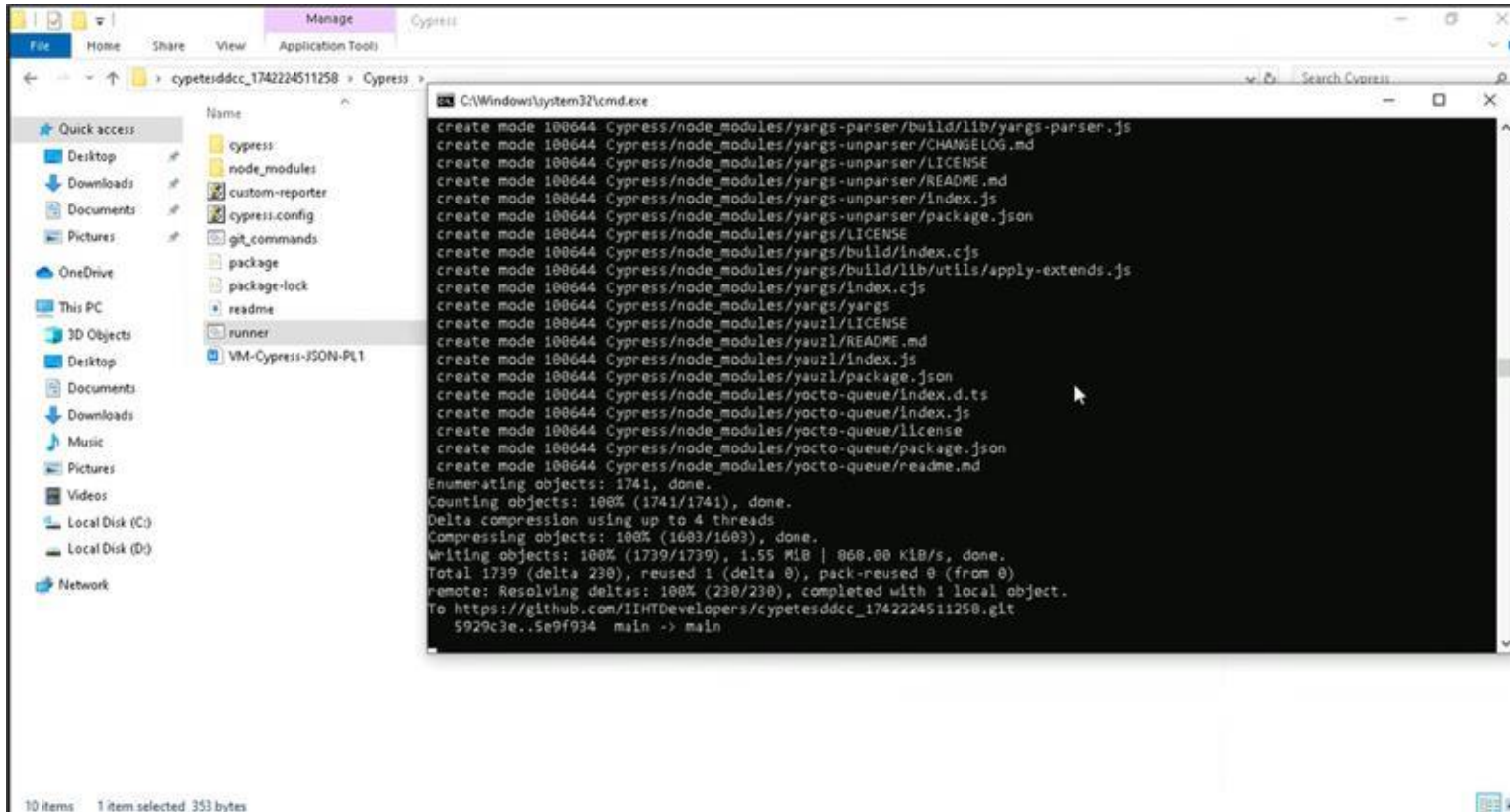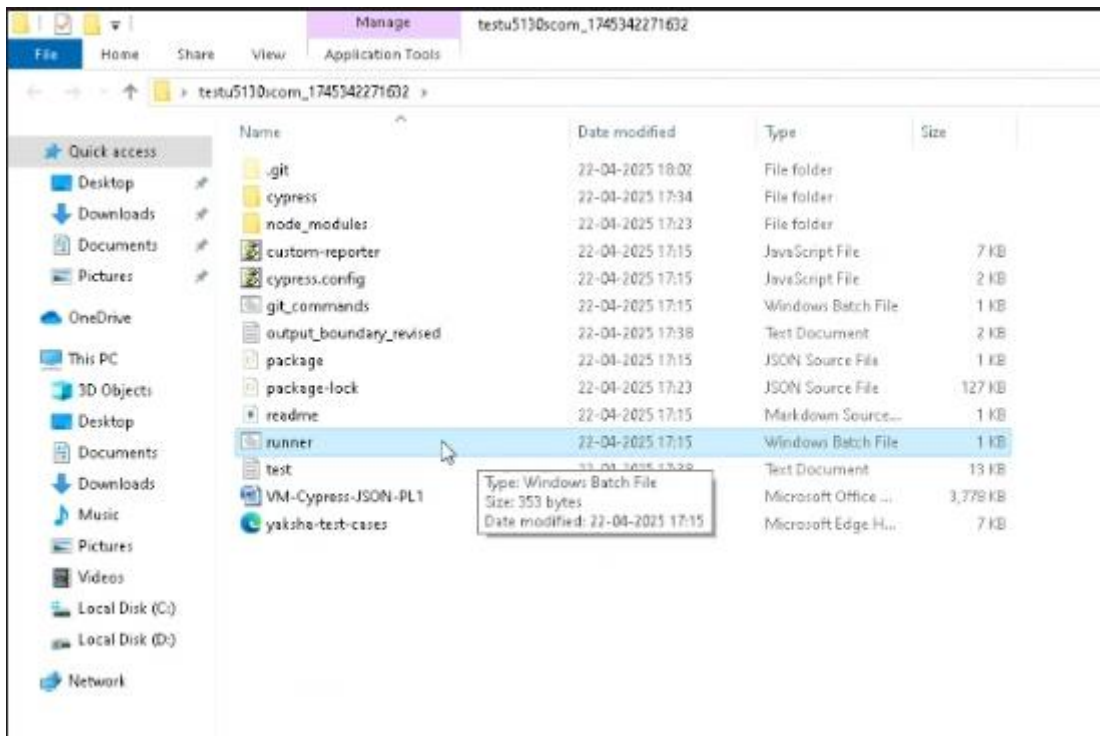Mymedic automation using cypress

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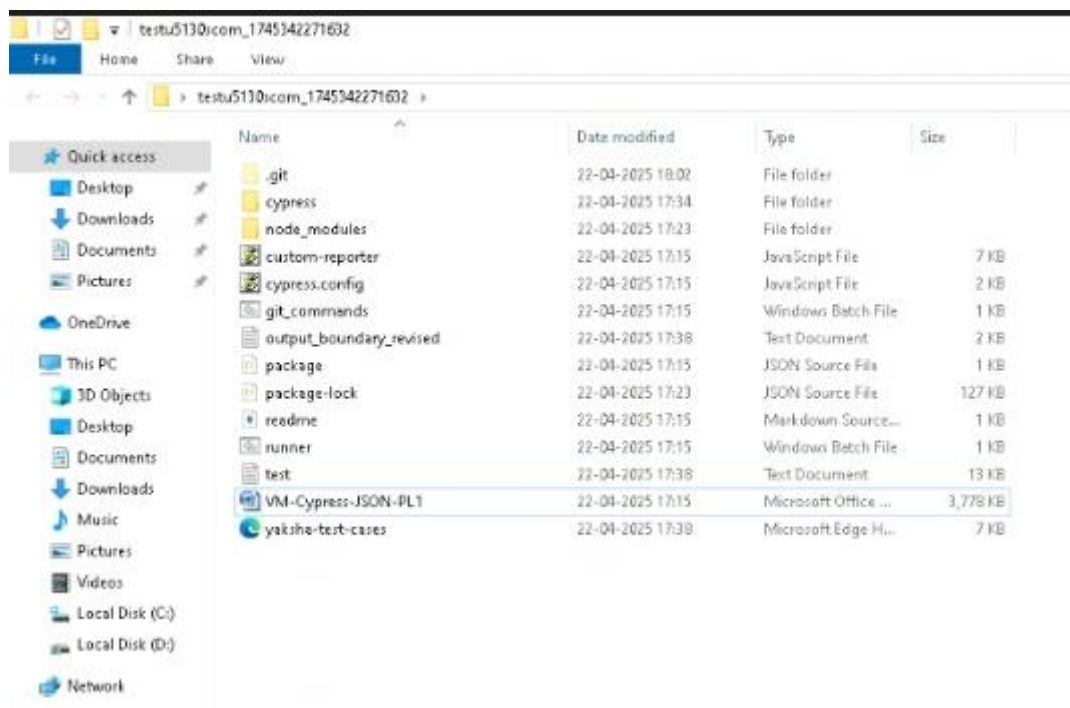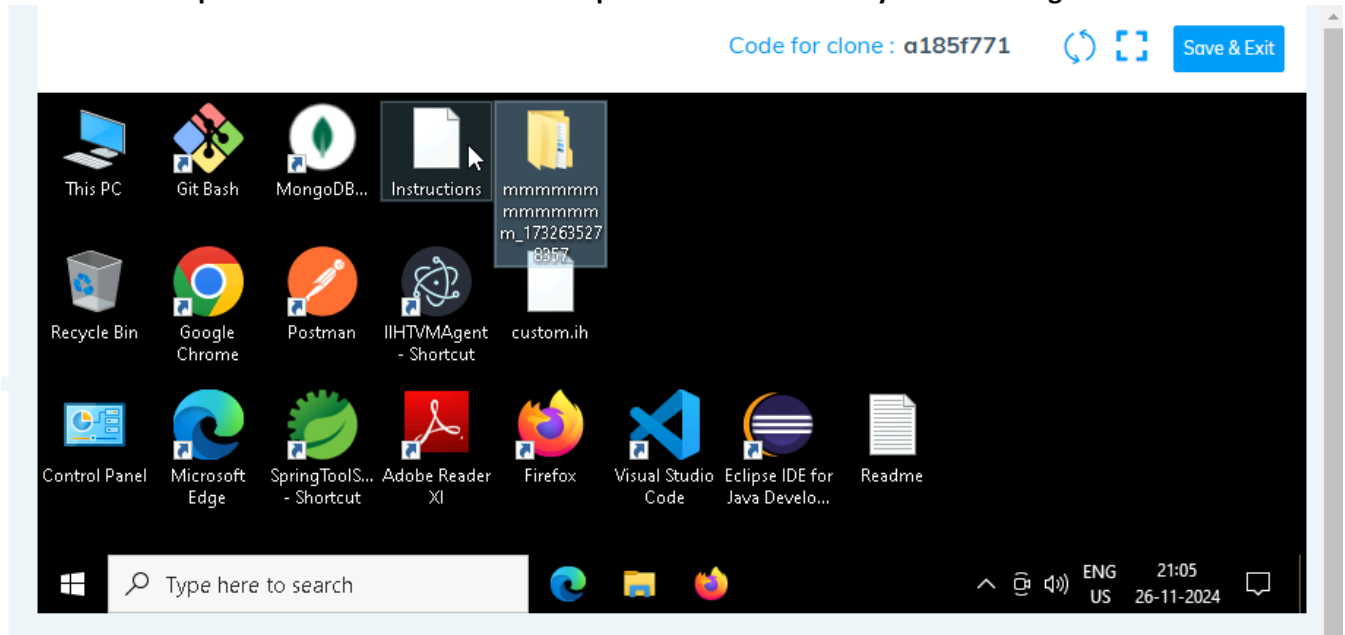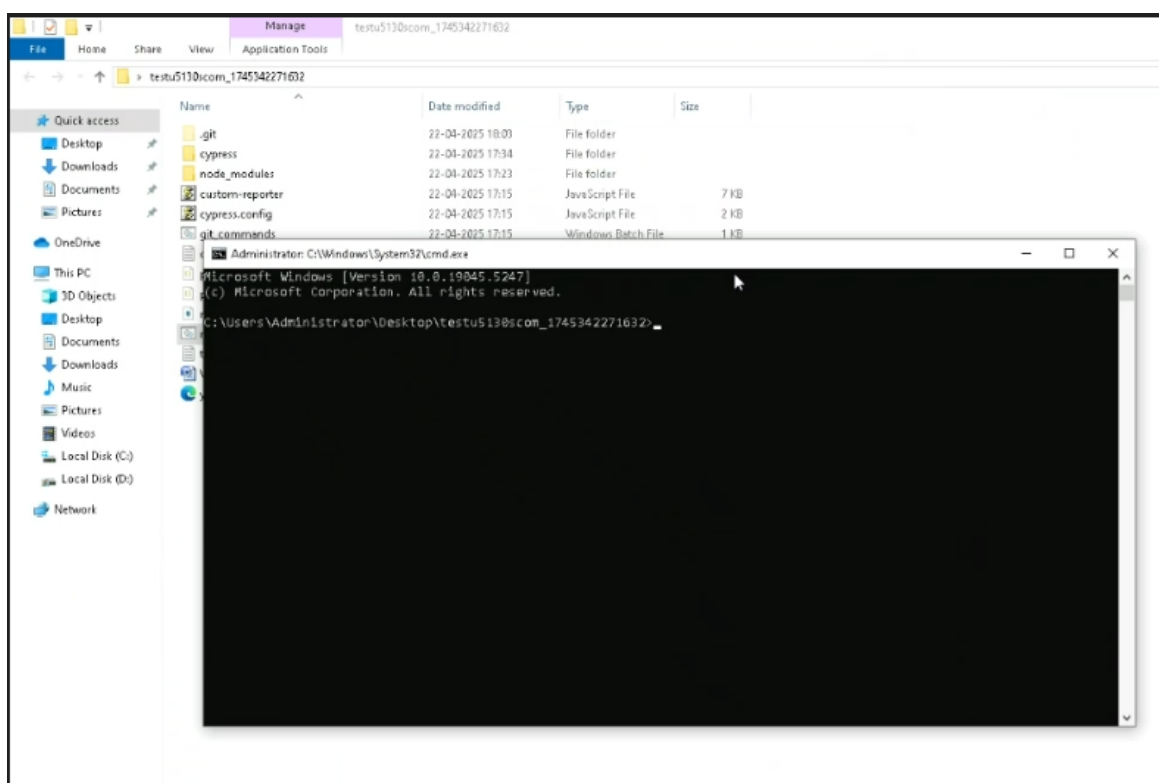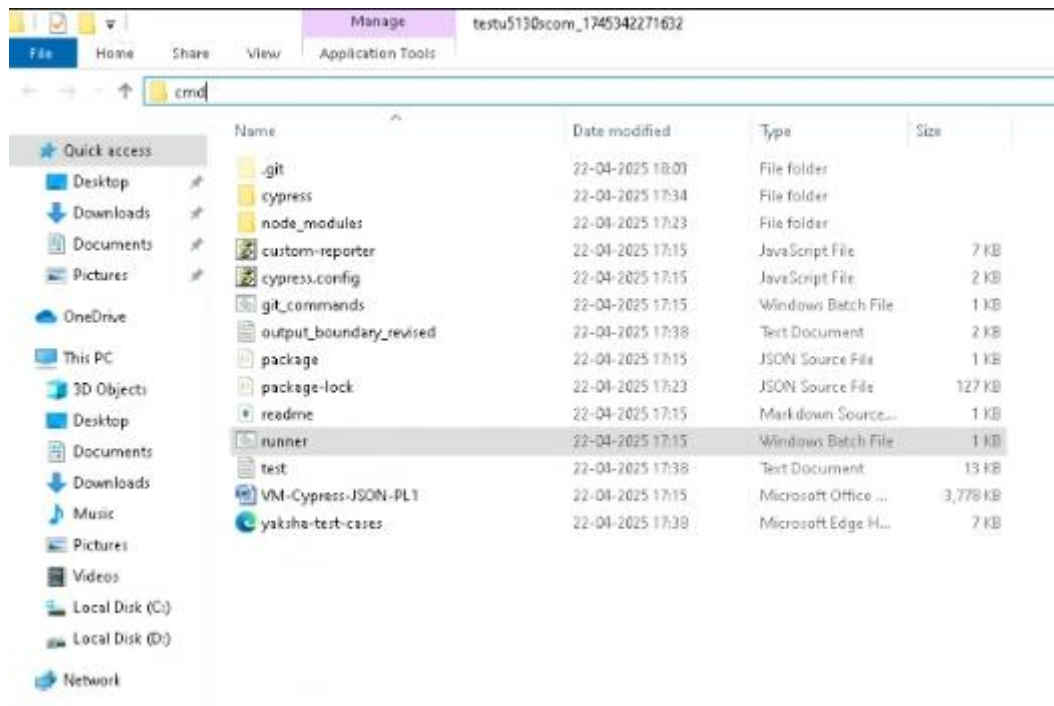

Mymedic automation using cypress

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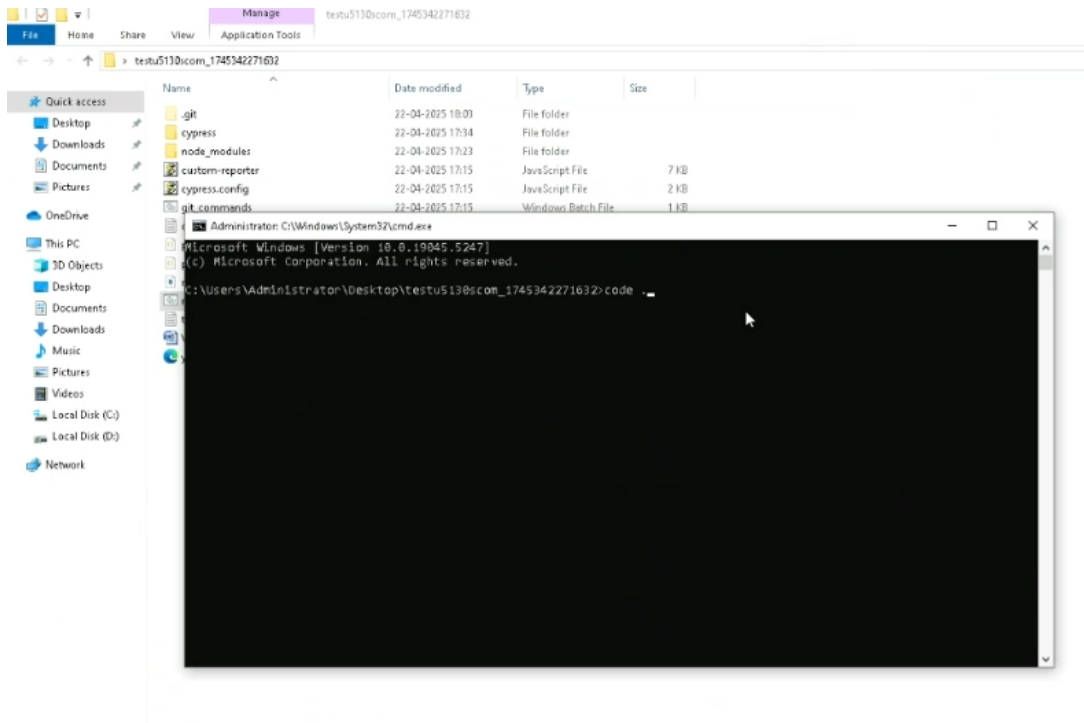

Mymedic automation using cypress

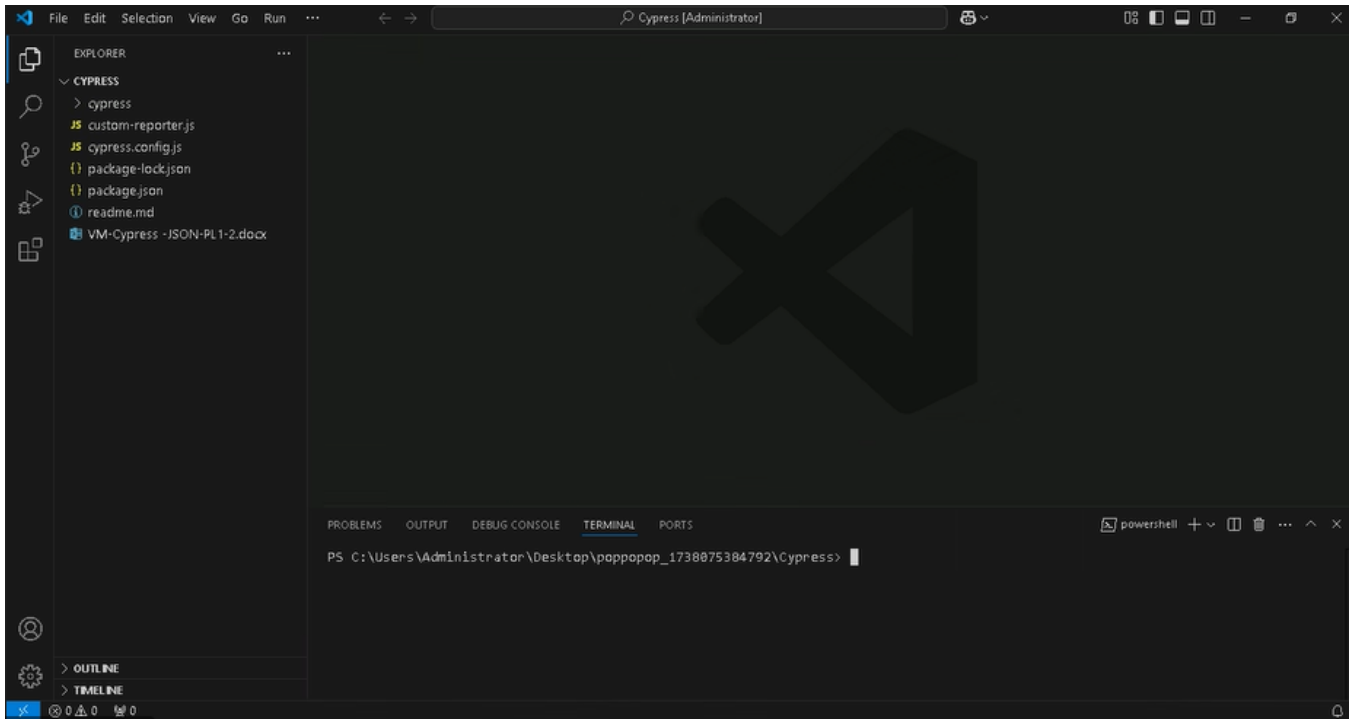**3. Open command prompt with its location using "cmd" in search bar**

Mymedic automation using cypress

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

   code .

5. Once VsCode is open:

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:**

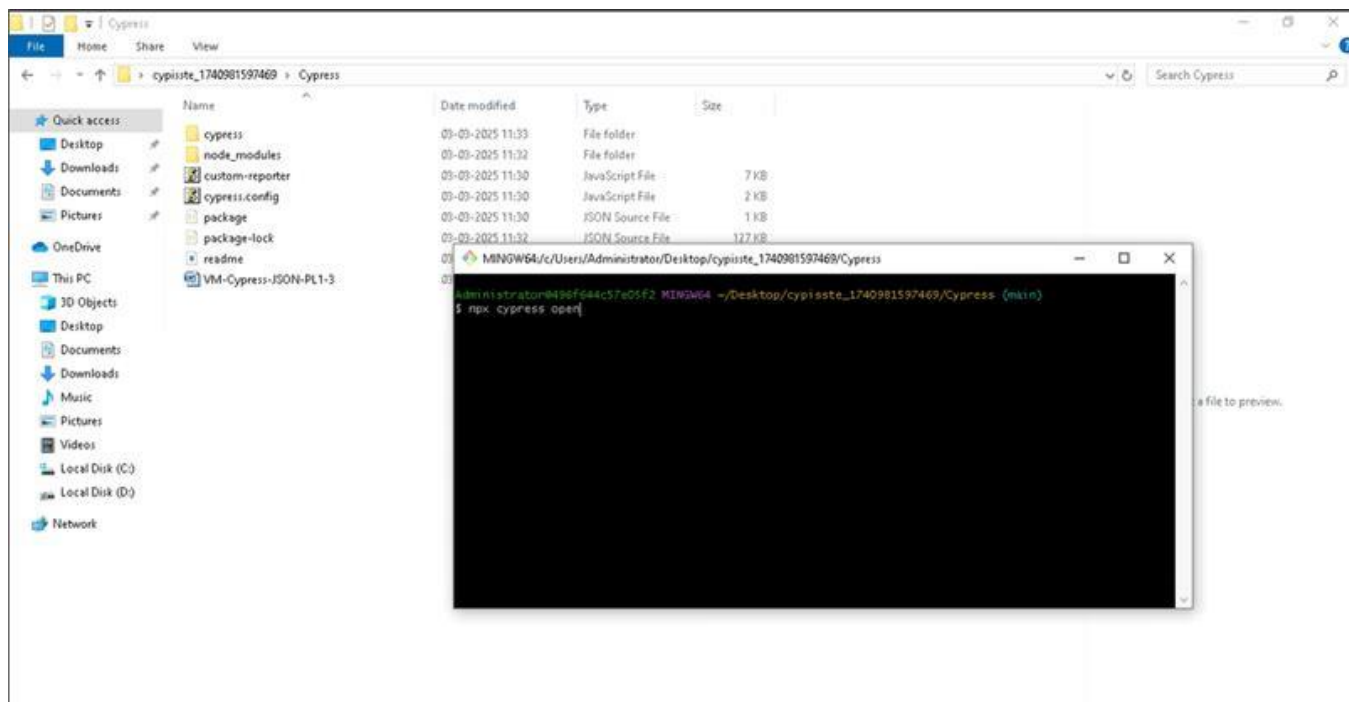   **npx cypress open**

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
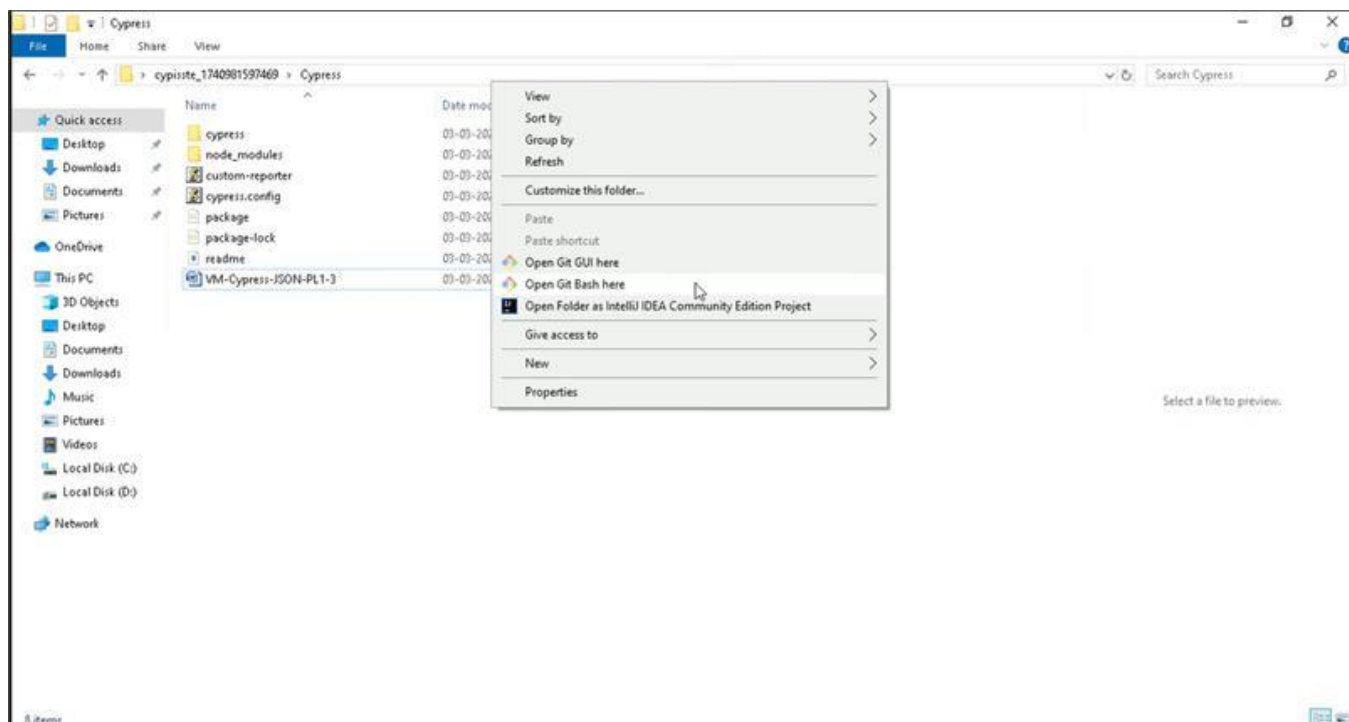
and then execute commands like:

Mymedic automation using cypress