

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 doctor.json, login.json, maternity.json, medicalRecord.json, patientName.json, pharmacy.json, radiology.json, settings.json and subStore.json.

Path	File	Description
\fixtures	doctor.json login.json maternity.json medicalRecord.json patientName.json pharmacy.json radiology.json settings.json	1. Contains data to read from json file.

Mymedic automation using cypress

	subStore.json	
PageObjects\Pages	<ul style="list-style-type: none"> ● ADTPage ● DoctorsPage ● IncentivePage ● LoginPage ● MaternityPage ● MedicalRecordsPage ● PharmacyPage ● RadiologyPage ● SubStorePage 	<ol style="list-style-type: none"> 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. 4. 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
	Verify Login with Valid Credentials	<p>1.The application should read the data from login.json file and fetch the user's name and password.</p> <p>2. It should call the method performLogin()</p> <p>3. Perform login method will perform authentication with the username and password.</p> <p>4. Verify admin name is visible on the home page.</p> <p>It is must to implement this login functionality at first and then implement any other test case.</p>	<p>Reference path</p> <p>\PageObjects\Pages\LoginPage</p> <p>methods</p> <p>performLogin()</p>	Successfully logs in with provided credentials. The user is logged in the admin page.
1	Handle Alert on Pharmacy	1. Navigate to "Pharmacy" module and select "Order" tab.	<p>Reference path</p> <p>\PageObjects\Pages\Pharmacy</p>	Handles alerts during the Good Receipt print process, and ensures the modal is

Mymedic automation using cypress

Test Case No.	Test Case Name	Test Steps to be performed	Path & Method Used	Expected Result
	Module	2. Click on "Add New Good Receipt" button. 3. Without adding any details, click on "Print Receipt" button. 4. 2 alert boxes should appear with messages as "Please select supplier" and "Please enter Invoice no."	yPage methods handlingAlertOnRadiology()	visible before performing further actions.
2	Verify to get the validation message when click on "Print Receipt" without filling any details	1. Navigate to "Pharmacy" module and select "Order" tab. 2. Click on "Add New Good Receipt" button. 3. Click on "Add New Item" button and fill details like "Item Name", "Batch no", "Item Qty" and "Rate" with data read from pharmacy.json file. 4. Add details to "Supplier Name" and "Invoice" as data read data from pharmacy.json file and any 3-digit random number like "777" respectively. 5. Click on "Print Receipt" button. 6. Verify the success message in pop-up at bottom right side as "Goods Receipt is Generated and Saved."	Reference path \PageObjects\Pages\PharmacyPage methods verifyPrintReceipt()	Verify success message popup - "Goods Receipt is Generated and Saved."
3	Verify to data range by select "Last 3 months" option from drop down	1. Use verifyDataWithinLastThreeMonths(). 2. Navigate to "Radiology" module. 3. Select "List Requests" tab. 4. Click on "-" button (present at left side of "OK" button) and select "Last 3 Months" option. 5. Click "OK" button.	Reference path \PageObjects\Pages\RadiologyPage methods verifyDataWithinLastThreeMonths()	Data should be present as per the selected date range using dropdown The 'Requested on' column date must fall within the "Last 3 months".
4	Verify that entering a keyword matching existing records in	1. Navigate to "Medical Records" module. 2. Select "MR Outpatient List" tab. 3. Enter data to be read from medicalRecord.json file in	Reference path \PageObjects\Pages\MedicalRecordsPage methods	Data should be present for the search, and the 'Gender' column should contain only patients in the "Female" category.

Test Case No.	Test Case Name	Test Steps to be performed	Path & Method Used	Expected Result
	the search bar returns the corresponding data.	"From" field. 4. Click on "Ok" button. 5. Enter gender data to be read from medicalRecord.json file in search field.	keywordMatching()	
5	Login with Invalid Credentials	1. Reset state by logging out if already logged in. 2. Use performLoginWithInvalidCredentials to check the invalid user credentials. 3. Invalid credentials should be read from login.json file. 4. Capture and verify error message: "Invalid User".	Reference path \PageObjects\Pages \ LoginPage methods performLoginWithInvalidCredentials()	Displays error message: "Invalid User". Logs success if message matches; logs failure otherwise.
6	Capture screenshot of Inventory Requisition section	1. Navigate to "SubStore" module. 2. Click on "Accounts" button. 3. Select "Inventory" tab and then "Inventory Requisition" sub tab. 4. Capture a screenshot of the page and save it in the "Screenshots" folder.	Reference path \PageObjects\Pages\SubStore Page methods captureInventoryRequisitionScreenshot()	Screenshot of the page is captured and saved successfully.
7	Verify to navigate to each section which are present in the "Inventory" sub module.	1. Navigate to "ADT" module. 2. Click on "Admitted Patients" tab. 3. Search for any patient name which data to be read from PatientName.json file. 4. Click on "..." button from table and select "Change Doctor". 5. Change doctor modal will open and then click on update button without filling any value.	Reference path \PageObjects\Pages\ADTPage methods verifyFieldLevelErrorMessage())	Verify a field level error message appears "Select doctor from the list."
8	Verify maternity allowance report is visible.	1. Navigate to "Maternity" module. 2. Click on "Reports" tab. 3. Click on "MaternityAllowance" button. 4. Enter date in "From" field to	Reference path \PageObjects\Pages\MaternityPage methods	Report should be visible when click on "show report" button and column with value is present.

Test Case No.	Test Case Name	Test Steps to be performed	Path & Method Used	Expected Result
		be read from maternity.json file and click on "Show Report" button. 5. Write code to check a column with name "Type" is present or not with value "Maternity Allowance".	verifyMaternityAllowanceReport()	
9	Verify imaging and lab order add successfully.	<ol style="list-style-type: none"> 1. Navigate to the "Doctor" module and then "In Patient Department" tab. 2. Click on the search bar and search patient with filed name as patientName read from doctor.json file. 3. Click on "Imaging" icon under the action column. 4. Select "imaging" option from drop down in "New Order" section. 5. Select searchOrderItem data read from doctor.json file in the search order item. 6. Click on "Proceed" button. 7. Click on "Sign" button. <p>Note :- If you cannot find the locator for "Proceed" button, please try to keep the page at 75%-80%.</p>	<p>Reference path</p> <p>\PageObjects\Pages\DoctorPage</p> <p>methods</p> <p>performInpatientImagingOrder()</p>	A success confirmation popup with the message: "Imaging and lab order add successfully" should appear.
10	Verify to filter the records by select "X-RAY" from Filter drop down.	<ol style="list-style-type: none"> 1. Navigate to "Radiology" module. 2. Read "filter" data from radiology.json file and select that option from "Filter" drop down. 3. Read "from" and "to" data from radiology.json file and put in "From" and add "To" fields. 4. Click on "Ok" button. 	<p>Reference path</p> <p>\PageObjects\Pages\RadiologyPage</p> <p>methods</p> <p>filterListRequestsByDateAndType()</p>	Record should filter out as per status.

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

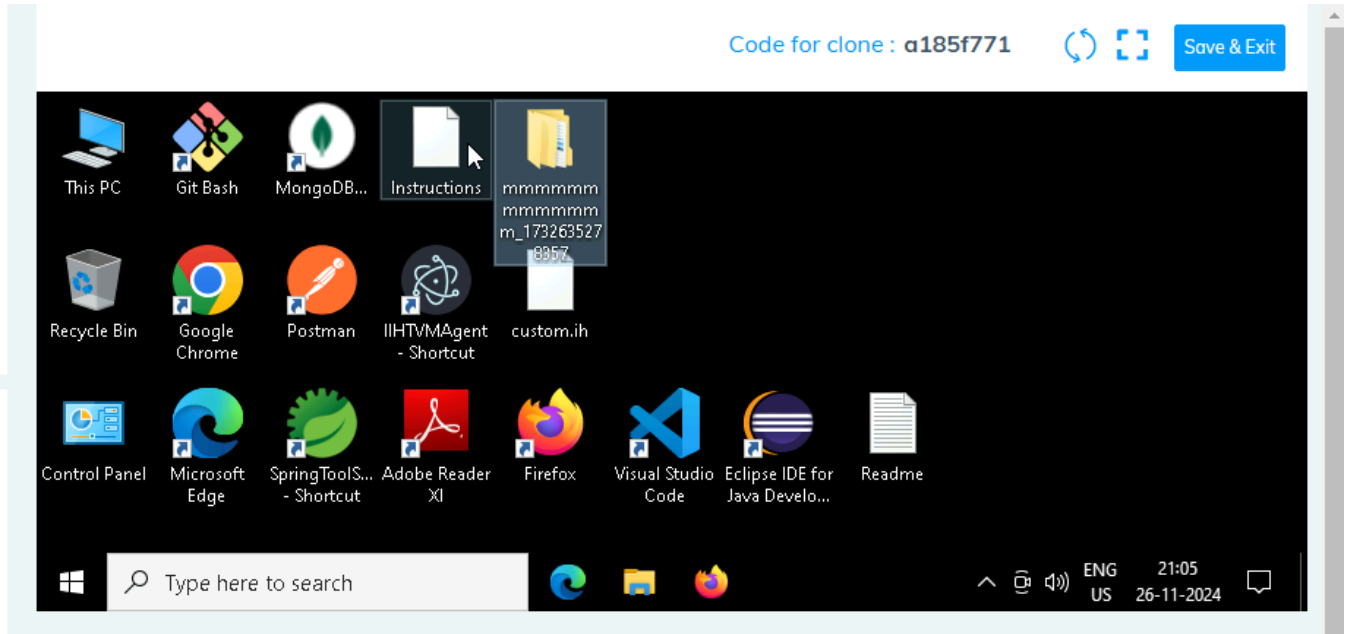
1.1 CODE QUALITY/OPTIMIZATIONS

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

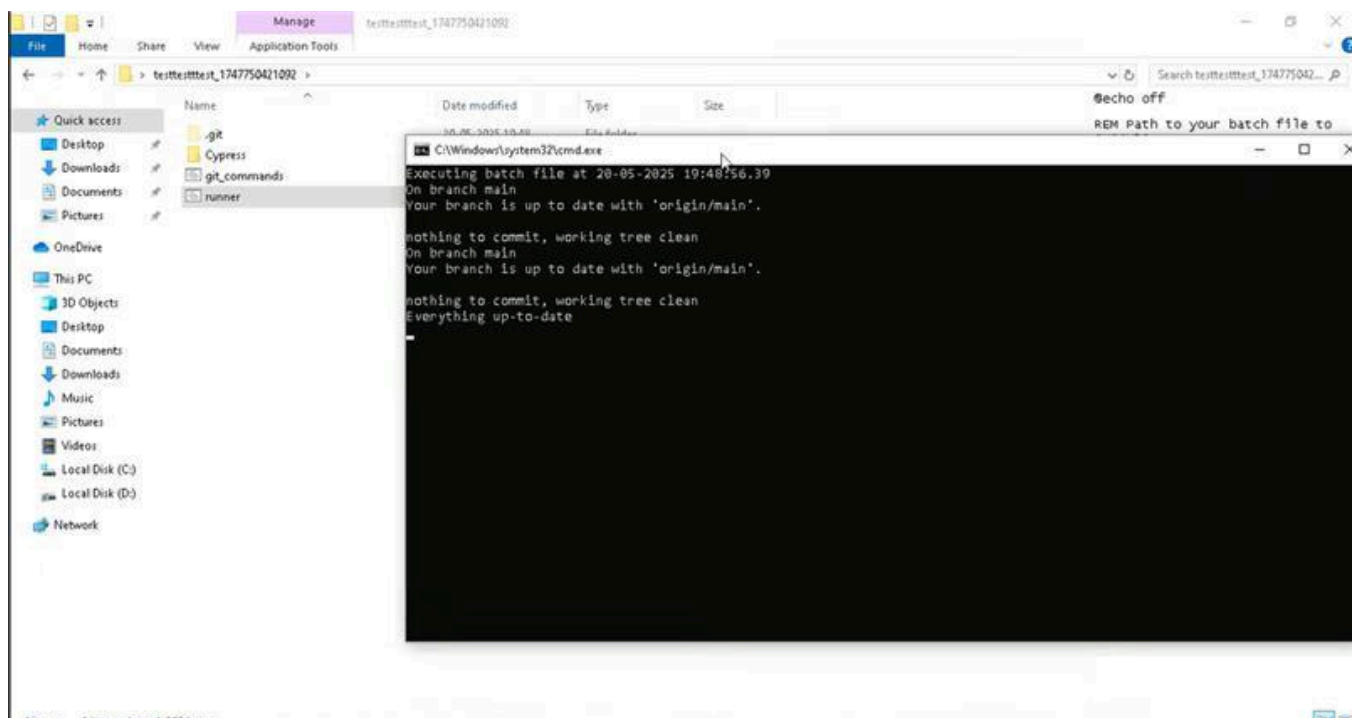
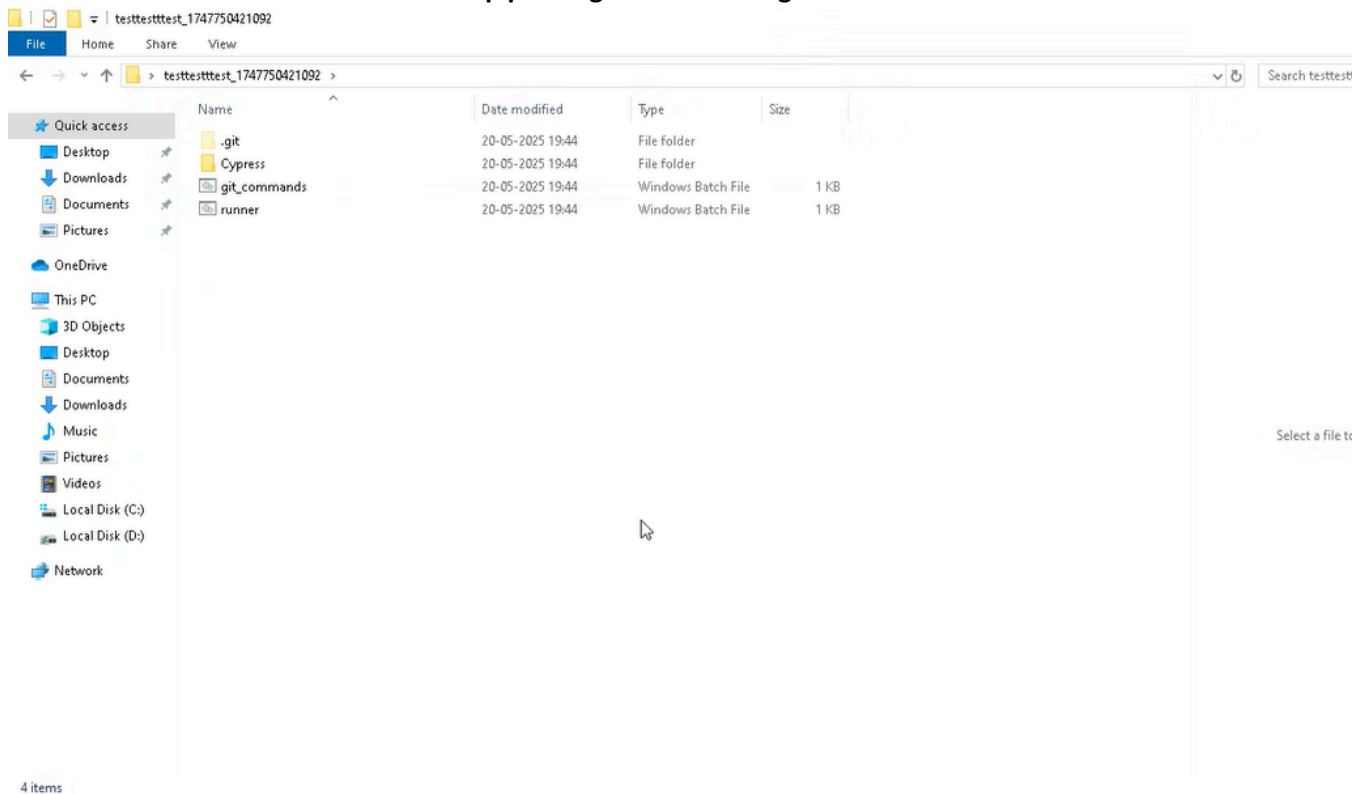
Execution Steps:

Steps for Execution:

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

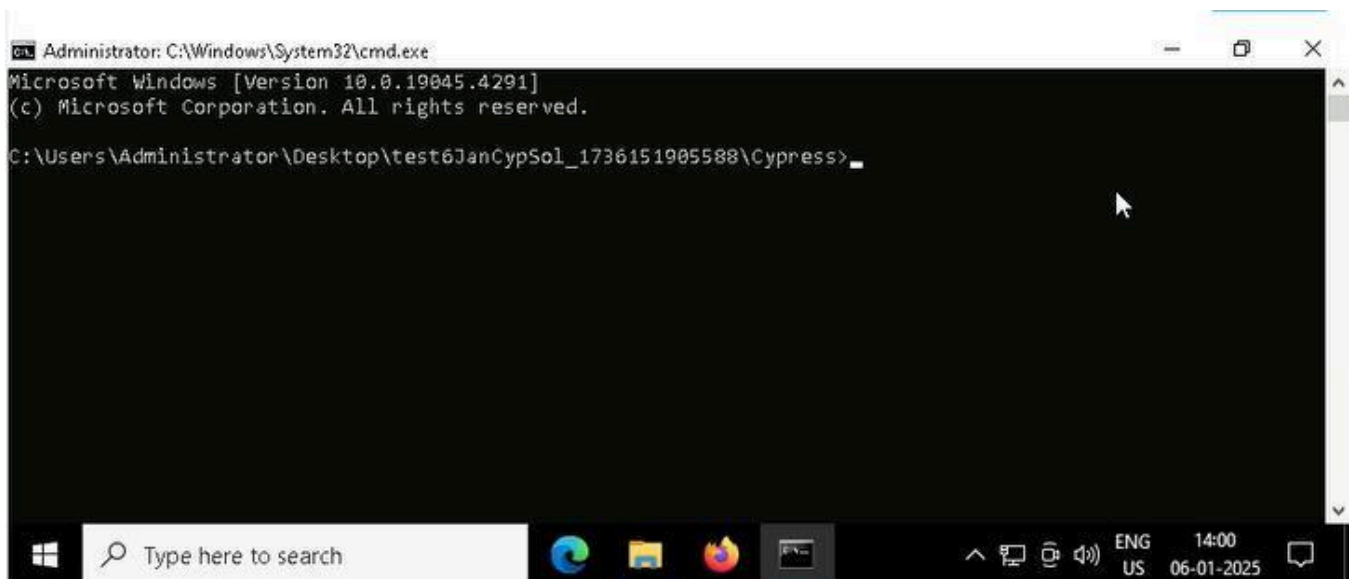
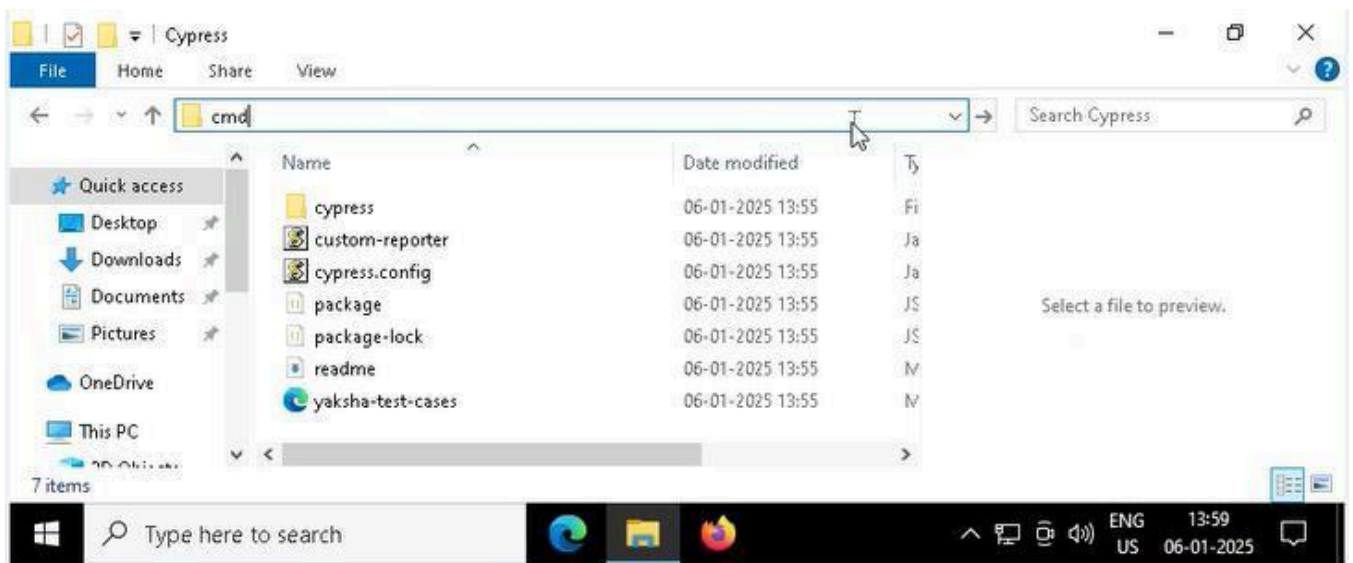


2. Execute the “runner” file to keep pushing the code at regular intervals.



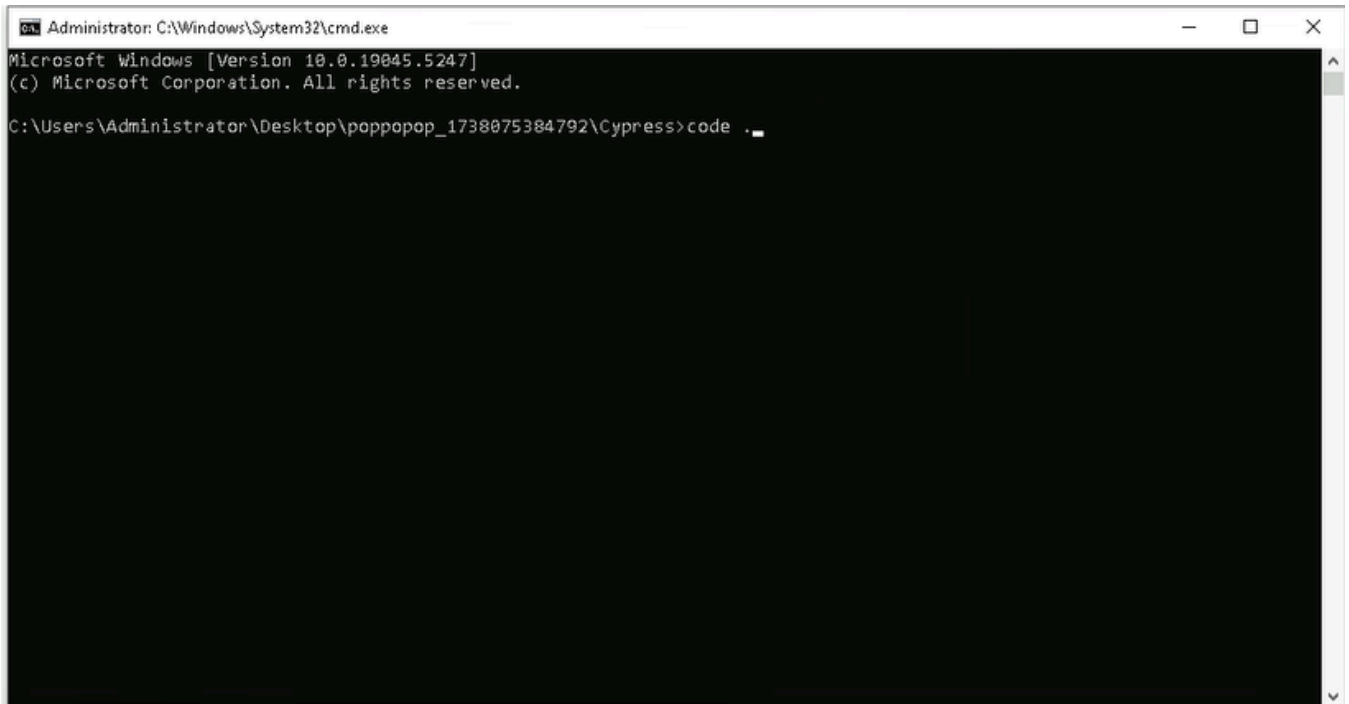
Mymedic automation using cypress

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



4. Open VS Code through cmd using below command:

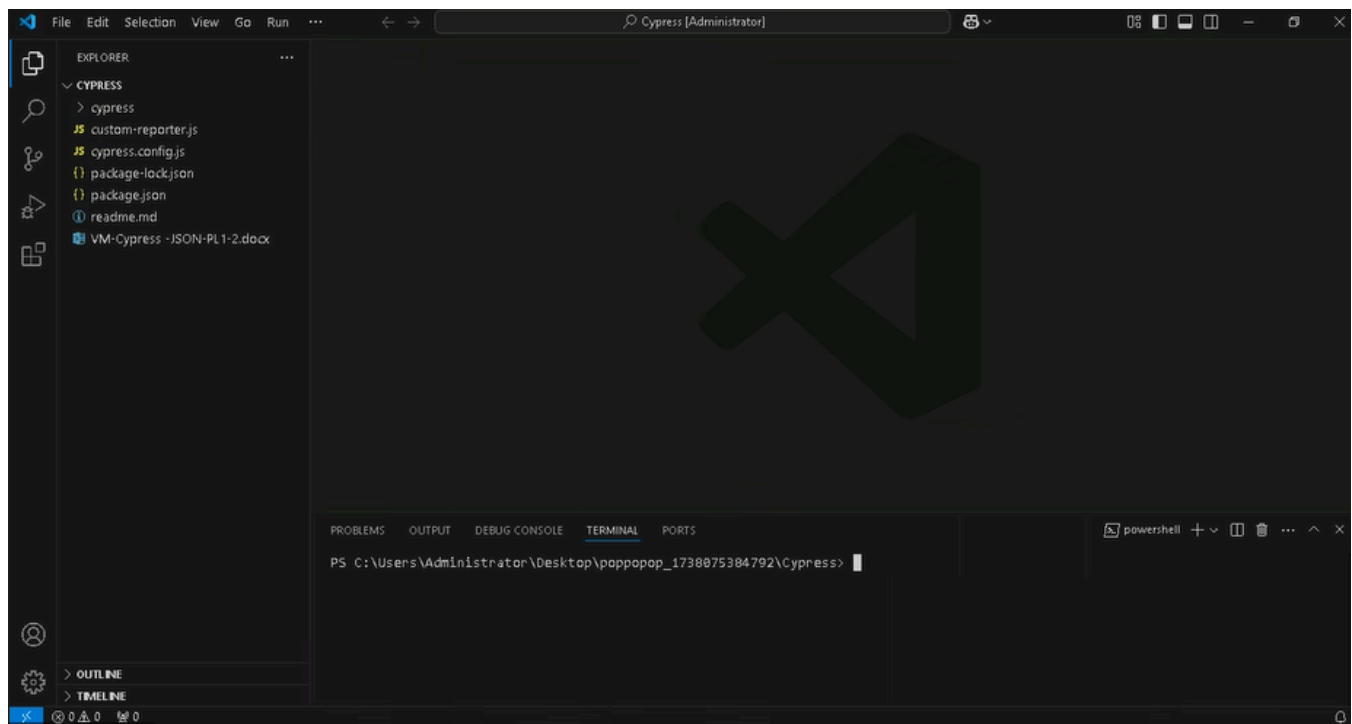
`code .`



```
Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5247]
(c) Microsoft Corporation. All rights reserved.

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

1. Once VsCode is open. Please open the terminal and execute below given next commands:



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

npm install

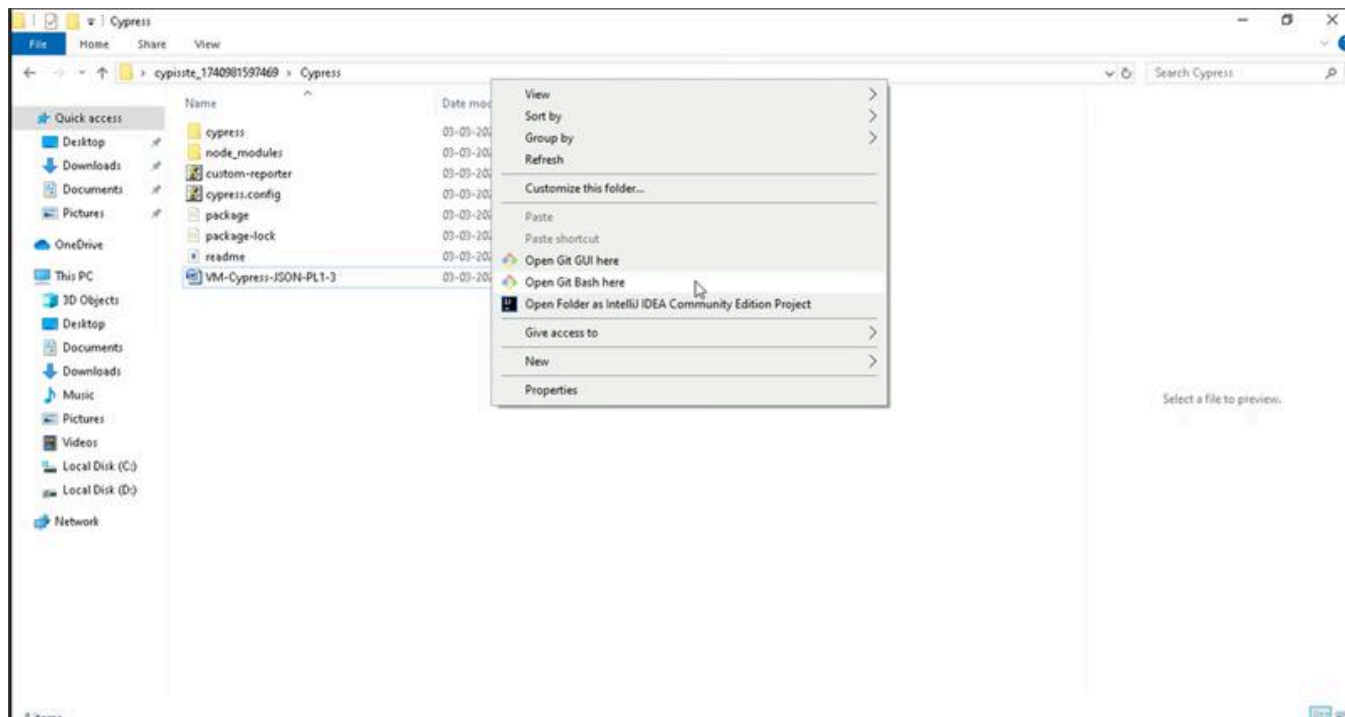
2. Run the following command to open the interactive Cypress Test Runner in the Cypress folder path:

npx cypress open

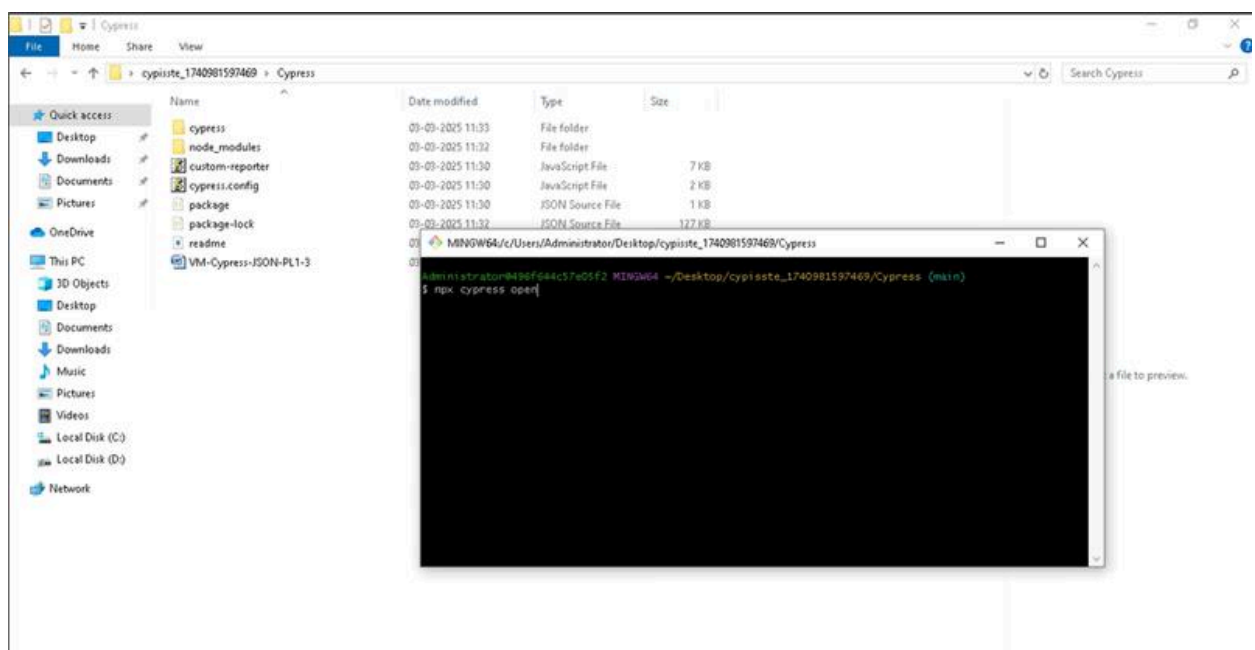
3. Run the following command to run test cases in headless mode in the Cypress folder path:

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:



Mymedic automation using cypress