

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 e2e/Data folder which you have to use like

PatientName.json, ValidLogin.json, appointment.json, dispensary.json, doctor.json, lab.json, maternity.json, medicalRecord.json, nursing.json, pharmacy.json, procurement.json, radiology.json, settings.json, subStore.json and testData.json.

Path	File	Description
/e2e/Data	<ul style="list-style-type: none">• PatientName.json• ValidLogin.json• appointment.json• dispensary.json• doctor.json• lab.json	1. Contains data to read from json file.

Mymedic automation using cypress

	<ul style="list-style-type: none"> ● maternity.json ● medicalRecord.json ● nursing.json ● pharmacy.json ● procurement.json ● radiology.json ● settings.json ● subStore.json ● testData.json 	
PageObjects/Pages	<ul style="list-style-type: none"> ● AccountingPage ● AdtPage ● CustomPage ● DispensaryPage ● LabPage ● LoginPage ● MaternityPage ● NursingPage ● PharmacyPage ● ProcurementPage ● SettingsPage ● SubStorePage ● UtilitiesPage 	<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	<ol style="list-style-type: none"> 1.The application should read the data from login.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. 	<p>Reference path</p> <p>\src\Pages\LoginPage</p> <p>methods</p> <p>performLogin()</p>	Successfully logs in with provided credentials. The user is logged in the admin page.

Mymedic automation using cypress

Test Case No.	Test Case Name	Test Steps to be performed	Path & Method Used	Expected Result
		It is must to implement this login functionality at first and then implement any other test case.		
1	Verify the main dispensary tooltip text	1. Navigate to 2.	Reference path \src\Pages\ methods -	A
2	Verify Navigation to Patient Overview from Past Days Records	1. Navigate to https://healthapp.yaksha.com/Home/Index#/Nursing/OutPatient 2. Click on the Past Days tab. 3. Enter the From Date as 01-01-2020 and click the OK button. 4. Search for the patient "Deepika Rani" in the list. 5. Locate the patient's record and click on Overview from the Actions column.	Reference path \src\Pages\NursingPage methods verifyPatientOverviewNavigationFromPastRecords()	Clicking on Overview should navigate the user to the Patient Overview page successfully.
3	Verify File Upload for a Past Patient Record	1. Navigate to https://healthapp.yaksha.com/Home/Index#/Nursing/OutPatient 2. Click on the Past Days tab. 3. Enter the From Date as 01-01-2020 and click the OK button. 4. Search for the patient "Deepika Rani" in the list. 5. Locate the patient's record and click on "Upload files" from the Actions column. 6. Verify that the Upload Files modal opens. 7. Select the Department as "Pathology". 8. Upload an image file. 9. Click on the Submit button.	Reference path \src\Pages\NursingPage methods verifyFileUpload()	A success confirmation popup with the message: "File Uploaded" should appear.
4	Verify Activation of BANK	1. Navigate to https://healthapp.yaksha.com/Home/Index#/Accounting/Transac	Reference path \src\Pages\AccountingPage	A success confirmation popup with the message:

Test Case No.	Test Case Name	Test Steps to be performed	Path & Method Used	Expected Result
	A/C # Ledger	tion/VoucherEntry 2. Click on the Settings tab. 3. Select Ledger from the available options. 4. Search for "BANK A/C #" in the ledger list. 5. Locate the ledger and click on Activate from the Actions column. 6. Accept the confirmation alert "Are you Sure want to Activate BANK A/C # ?".	methods verifyBankActivation()	"BANK A/C # is now activated." should appear.
5	Verify Deactivation of BANK A/C # Ledger	1. Navigate to https://healthapp.yaksha.com/Home/Index#/Accounting/Transaction/VoucherEntry 2. Click on the Settings tab. 3. Select Ledger from the available options. 4. Search for "BANK A/C #" in the ledger list. 5. Locate the ledger and click on Activate from the Actions column. 6. Accept the confirmation alert "Are you Sure want to Deactivate BANK A/C # ?".	Reference path \src\Pages\AccountingPage methods verifyBankDeactivation()	A success confirmation popup with the message: "BANK A/C # is now Deactivated.." should appear.
6	Verify all sub-modules are displayed correctly after Clicking on the "SubStore" Module.	1. Navigate to https://healthapp.yaksha.com/Home/Index#/WardSupply 3. "Select your Substore" pop up 4. Click on "Account" sub-modal 5. Click on the "Inventory" sub-module 6. Click on "Pharmacy" sub-module	Reference path \src\Pages\SubStorePage methods verifySubModulesVisible()	All sub-modules should be displayed correctly. Expected Sub modules are : Pharmacy, Inventory
7	Verify the tooltip and it's text present on hover the mouse on "Star"	Pre condition: User should be logged in and it is on Maternity module 1. Navigate to "Maternity list" sub-module 2. Hover the mouse in star	Reference path \src\Pages\MaternityPage methods verifyTooltipTextIsVisible()	Tooltip should be present when hover the mouse on star with text "Remember this date"

Test Case No.	Test Case Name	Test Steps to be performed	Path & Method Used	Expected Result
8	Verify Request for Quotation Generation	<ol style="list-style-type: none"> 1. Navigate to https://healthapp.yaksha.com/Home/Index#/ProcurementMain/PurchaseRequest/PurchaseRequestList 2. Click on the Quotation tab. 3. Click on the Request for Quotation button. 4. Enter the following data in the respective fields: Subject: test subject Description: test description Select Vendor: Ashar & Company Item Name: Soap Quantity: 2 5. Click on the Request button. 	<p>Reference path</p> <p>\src\Pages\ProcurementPage</p> <p>methods</p> <p>verifyRequestForQuotationGeneration()</p>	A success confirmation popup with the message: "Request For Quotation is Generated and Saved" should appear.
9	Verify table filtering for "Male Ward"	<ol style="list-style-type: none"> 1. Navigate to https://healthapp.yaksha.com/Home/Index#/Lab/Dashboard 2. Select the Sample Collections tab. 3. Enter From Date as 01-01-2020 and click OK. 4. Hover over the Requesting Department column and click Hamburger Menu. 5. Select Starts with from the dropdown. 6. Enter Male Ward in the text field. 	<p>Reference path</p> <p>\src\Pages\LabPage</p> <p>methods</p> <p>verifyTableFilteringForMaleWard()</p>	Table should display only rows with "Male Ward" in the Item column.
10	Verify to export the order section data	<ol style="list-style-type: none"> 1. Navigate to https://healthapp.yaksha.com/Home/Index#/Pharmacy/Dashboard 2. Click on order section 3. Click on "Export" button 	<p>Reference path</p> <p>\src\Pages\PharmacyPage</p> <p>methods</p> <p>verifyExportAndDownload()</p>	User should able to export the order section data
11	Verify Warning Popup for Mandatory Fields in	<ol style="list-style-type: none"> 1. Navigate to the Utilities module and select the "Scheme Refund" tab. 2. If required, please select any counter value and then select the "Scheme Refund" tab. 	<p>Reference path</p> <p>\src\Pages\UtilitiesPage</p> <p>methods</p> <p>verifyWarningPopupForMandatoryFields()</p>	A warning popup should appear with the message: "Please fill all the mandatory fields."

Test Case No.	Test Case Name	Test Steps to be performed	Path & Method Used	Expected Result
	Scheme Refund	3. Click on the "New scheme Refund Entry" button. Now click on save without entering value in any field.		
12	Verify Price Category Enable/Disable	1. Navigate to the "Settings" module. 2. Click on more... and select the "Price Category" tab. 3. Click on the "Disable" button to disable any Code in the table. 4. Verify a success message appears with the message "Deactivated." Activate the same code by clicking the "Activate" button and verify the success message as "Activated".	Reference path \src\Pages\SettingsPage methods verifyDisablePriceCategory() verifyEnablePriceCategory()	A success message is displayed for both actions: "Deactivated." for disabling and "Activated." for enabling.
13	Verify Navigation Between Different Tabs	1. Navigate to the "SubStore" module. 2. Click on the "Accounts" button. 3. Select the "Inventory" tab. 4. Click and navigate between different tabs like "Stock", "Inventory Requisition", "Consumption", "Reports", "Patient Consumption" and "Return".	Reference path \src\Pages\SubStorePage methods verifyNavigationToSubStoreModule() navigateToAccounts() verifyNavigationToInventoryRequisition() verifyNavigationToConsumptions() verifyNavigationToReports() verifyNavigationToPatientConsumptions() verifyNavigationToReturn() verifyNavigationToStock()	Ensure that it should navigate to each sub tab of the "Inventory" tab.

Test Case No.	Test Case Name	Test Steps to be performed	Path & Method Used	Expected Result
14	Capture screenshot of Inventory Requisition section.	<ol style="list-style-type: none"> 1. SNavigate 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 \src\Pages\SubStorePage methods captureScreenshotOfInventoryRequisitionSection()	Screenshot of the page is captured and saved successfully.
15	Verify to navigate to each section which are present in the "Inventory" sub-module	<ol style="list-style-type: none"> 1. Navigate to ADT module. 2. Click on "Admitted Patients". 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 \src\Pages\ADTPage methods verifyInventorySubModuleNavigation()	Verify a field level error message appears "Select doctor from the list."

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

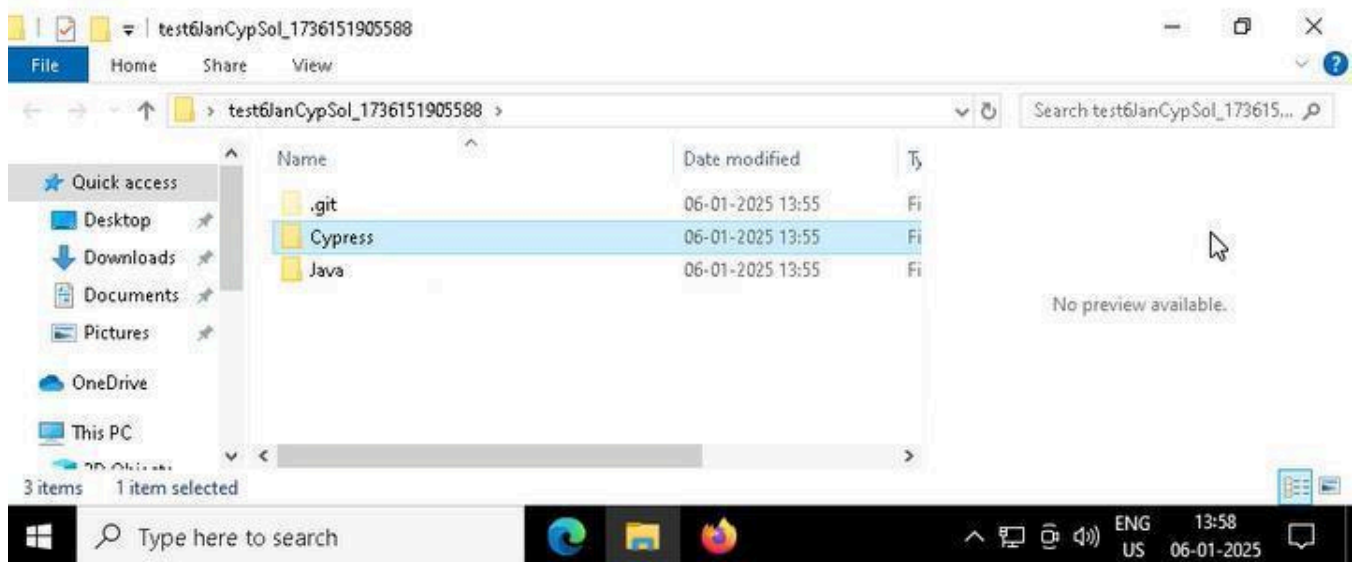
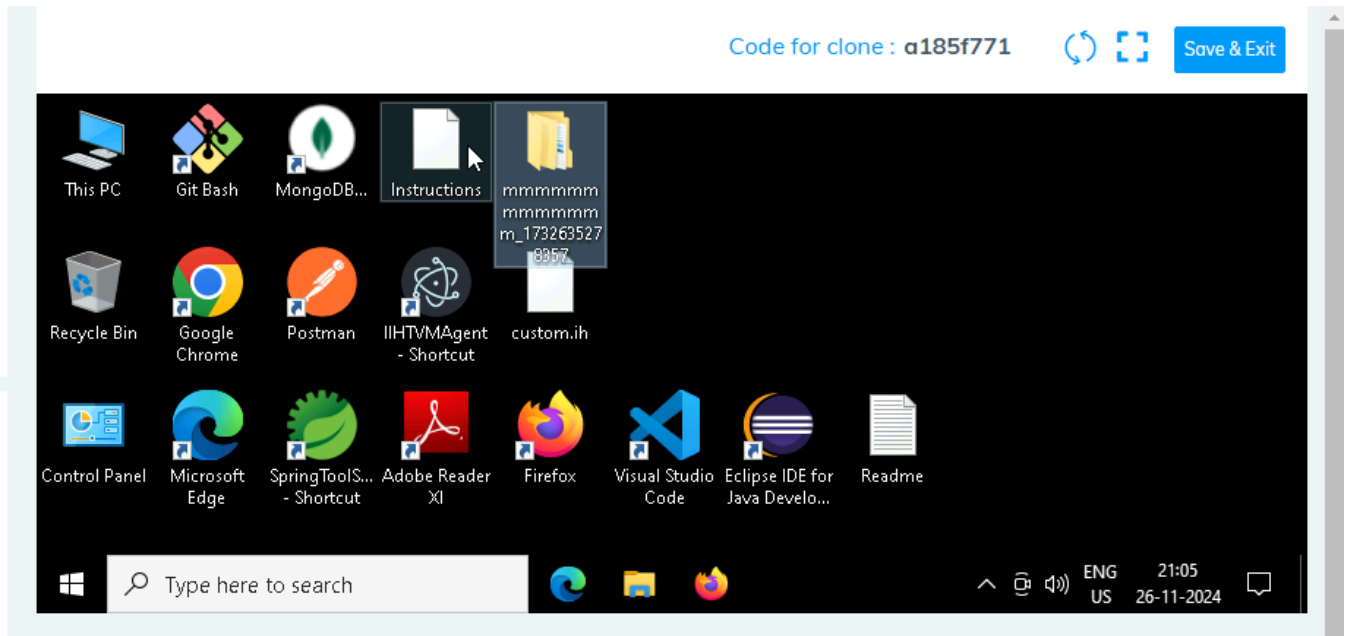
1. Associates should have written clean code that is readable.
2. Associates need to follow SOLID programming principles.

Mymedic automation using cypress

Execution Steps:

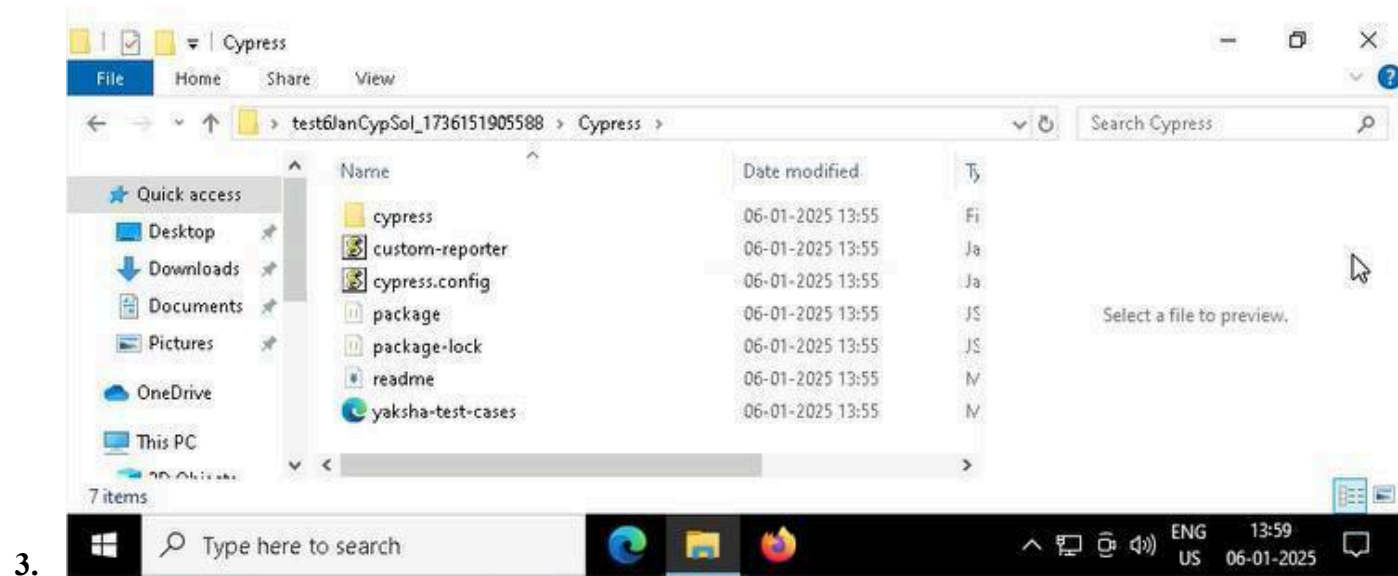
Steps for Execution:

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

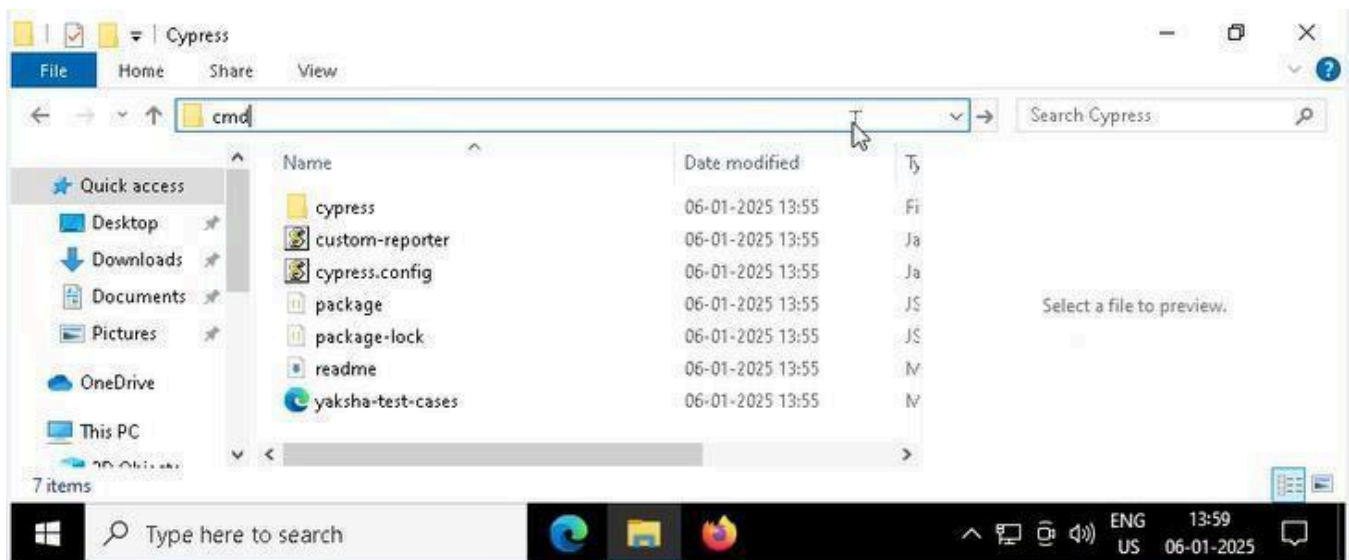


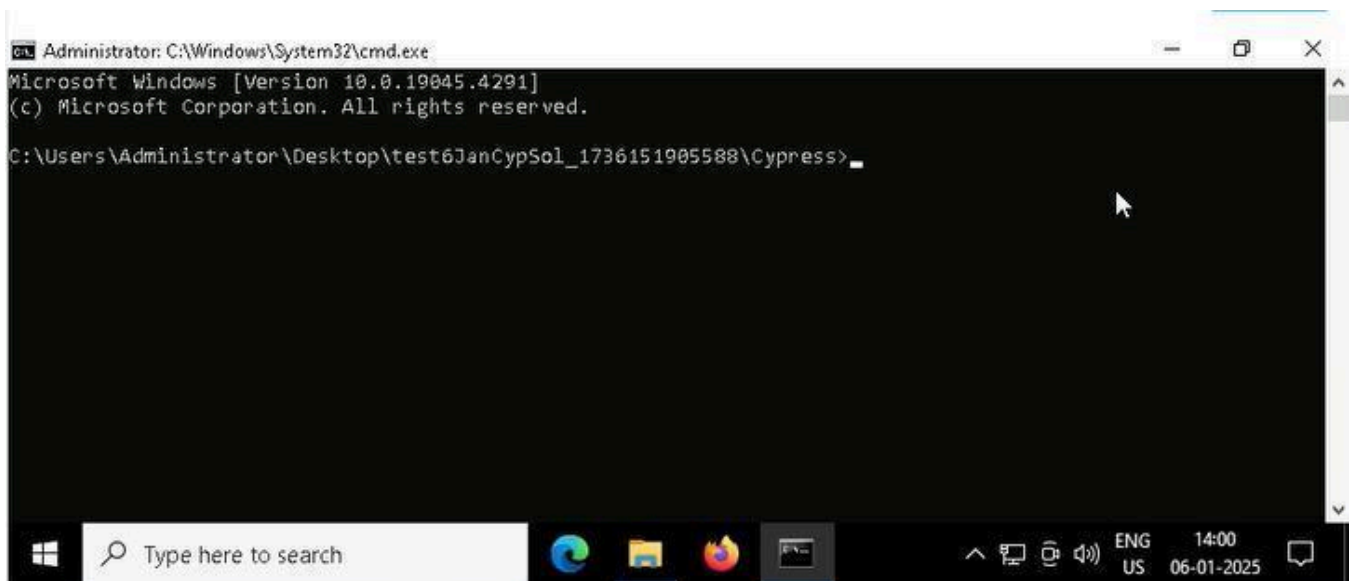
2. Go into the Cypress folder

Mymedic automation using cypress



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



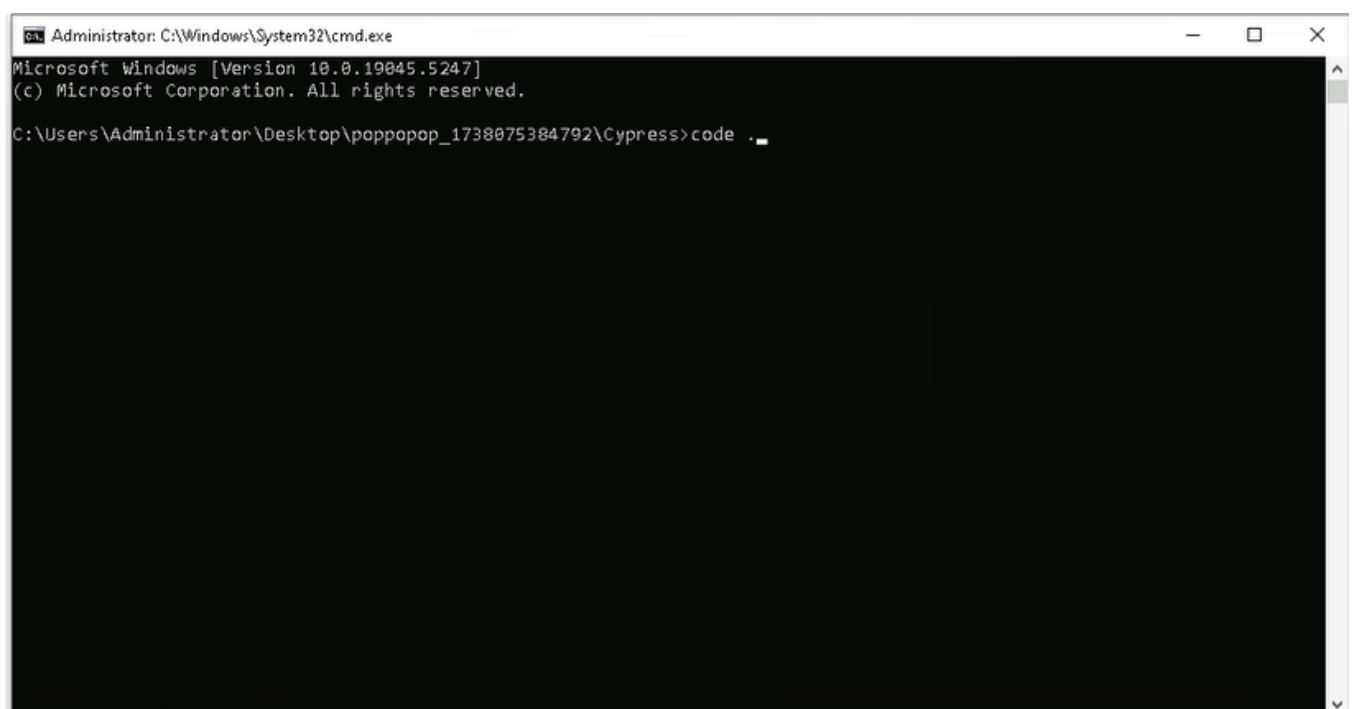


```
Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

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

5. Open VS Code through cmd using below command:

code .

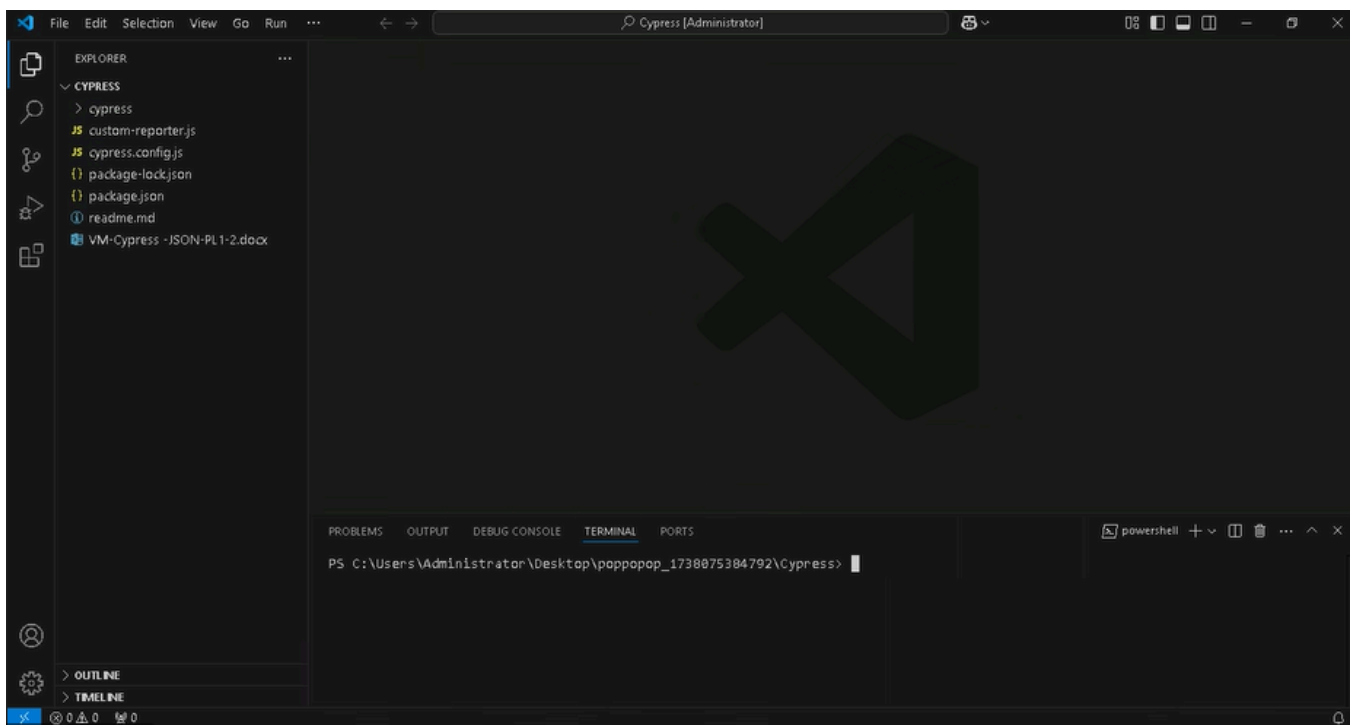


```
Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5247]
(c) Microsoft Corporation. All rights reserved.

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

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

Mymedic automation using 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:

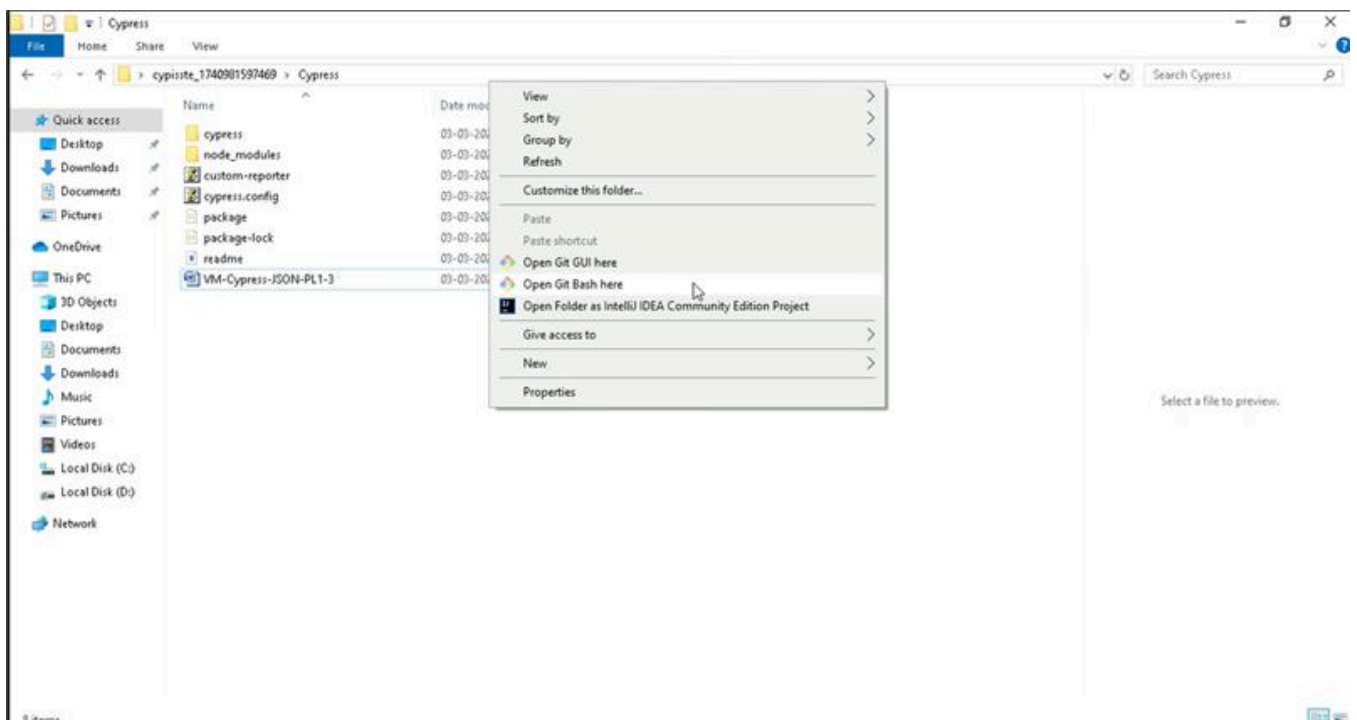
npx cypress open

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

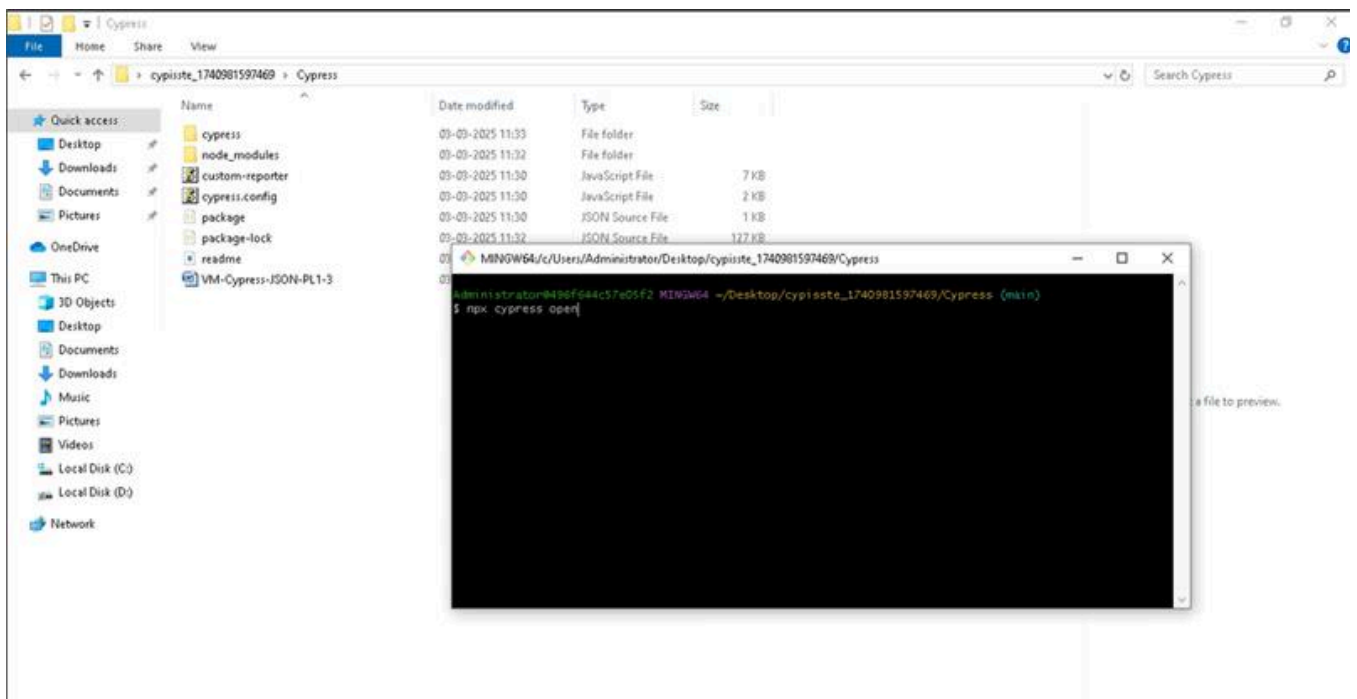
npx cypress run

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:

Mymedic automation using cypress



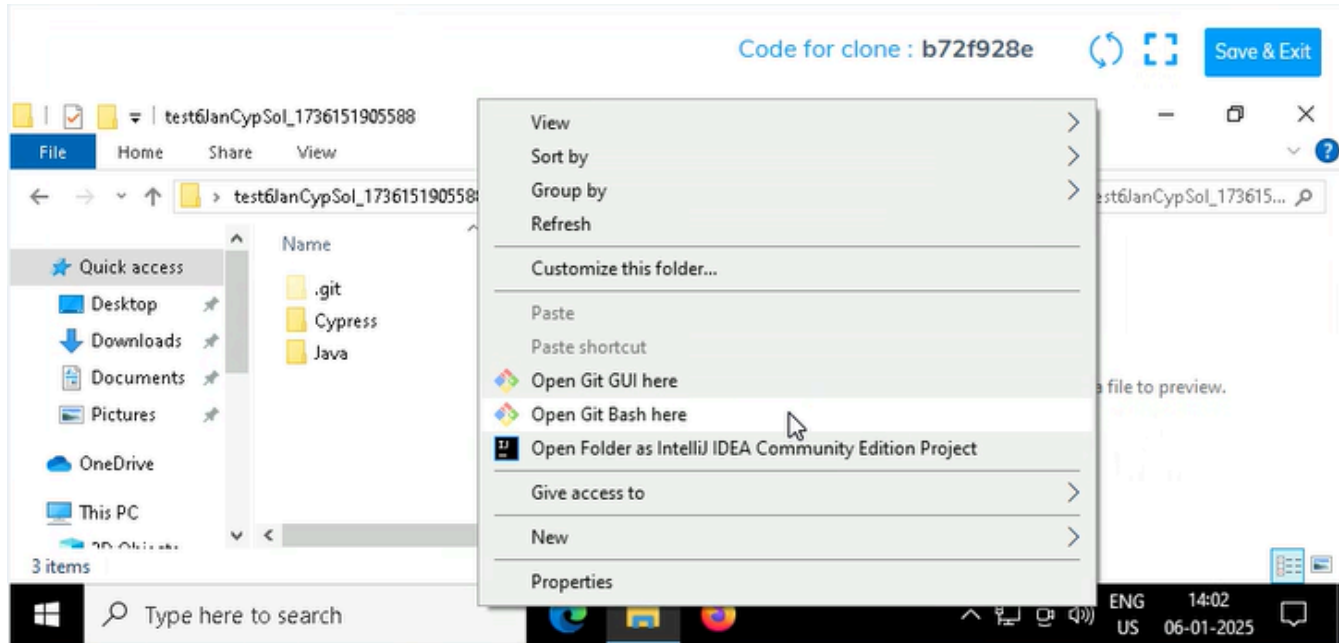
and then execute commands like:



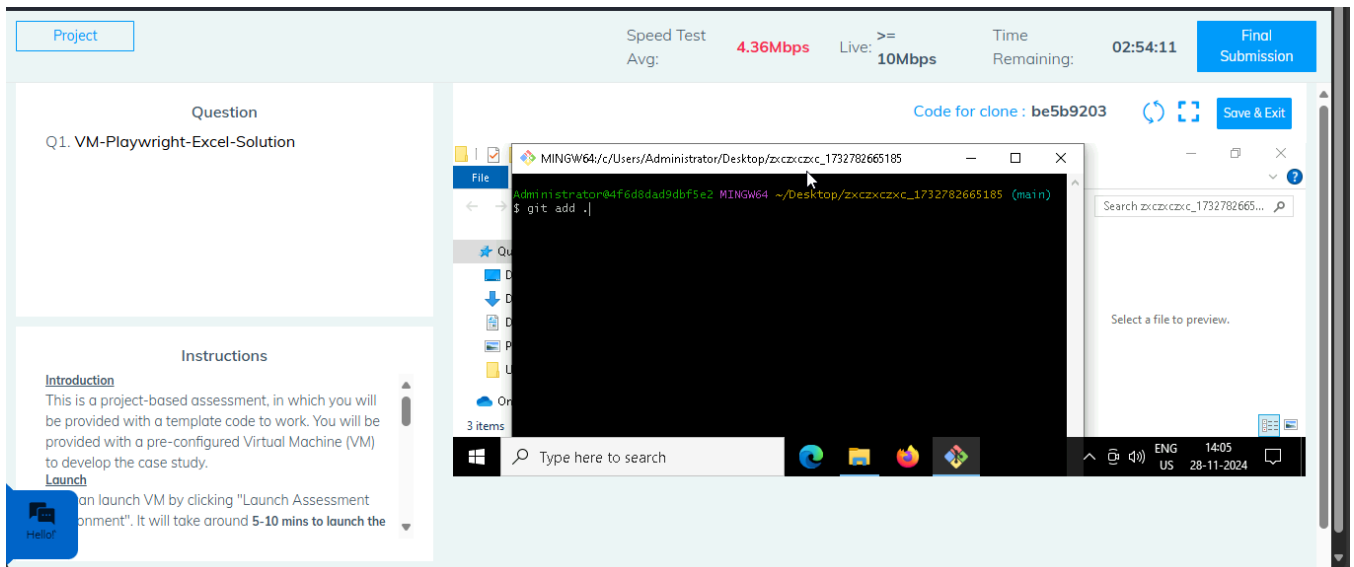
4. Once you have executed the test cases. Now it is necessary to push your code to git. For this, please go to Mymedic automation using cypress

inside the folder created on desktop with the email id you have used to login and then:

1. Open gitbash





2. Add all files



3. Commit the changes








Mymedic automation using cypress

Code for clone : be5b9203   Save & Exit



MINGW64/c/Users/Administrator/Desktop/zxczxczc_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/zxczxczc_1732782665185 (main)
\$ git commit -m "committing the updates"

 Type here to search      ENG 14:10
US 28-11-2024 








4. Push the changes

Code for clone : be5b9203   Save & Exit

MINGW64/c/Users/Administrator/Desktop/zxczxczc_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_externs.js

Administrator@4f6d8dad9dbf5e2 MINGW64 ~/Desktop/zxczxczc_1732782665185 (main)
\$ git push

 Type here to search      ENG 14:12
US 28-11-2024 

Mymedic automation using cypress

