YAKSHA HEALTH APP WITH JAVACRIPT 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 Data folder like PatientName.json and ValidLogin.json.

Path	File	Description
e2e\Data	PatientName.json ValidLogin.json	1. Contains data to read from json file.
PageObject\Pages	• ADTPage • AdminPage • AppointmentPage • DashboardPage • DoctorPage • IncentivePage	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

	● LoginPage ● OperationTheaterPage ● PatientPage ● ProcurementPage ● SettingsPage ● SubstorePage ● UtilitiesPage	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	1.The application should read the data from ValidLogin.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** \PageObject\Pages **LoginPage** **methods** performLogin()	Successfully logs in with provided credentials. The user is logged in the admin page.
1	Verify the presence of Visit Type drop down by selecting "New patient" option	1. Use verifyVisitTypeDropdown(). 2. Open "Appointment Booking List" tab inside Appointment module. 3. Select any counter, if it's not selected already and then select "Appointment Booking List" tab. 4. Select "New Patient" from "Visit Type" dropdown. 5. Select "All Doctors" from "Doctor" dropdown. 6. Pick "01-01-2024" in "From Date" field. 7. Pick "31-03-2024" in "To Date" field. 8. Click on "Show Patient" button.	**Reference path** \PageObject\Pages\AppointmentPage **methods** verifyVisitTypeDropdown()	The "Visit Type" column in list should contain only patients of "New" category.
2	Handle Alert for OT Booking Without Patient Selection	1. Navigate to Operation theatre page. 2. Use handleOtBookingAlert() to handle the alert. 3. Click on "New OT Booking" button. 4. Verify that the "Booking OT Schedule | New Patient" modal is displayed. 5. Without entering any details, within the modal, click on the "Add New OT" button. 6. Handle the alert with message "Patient not Selected! Please Select the patient first!".	**Reference path** \PageObject\Pages\OperationTheatrePage **methods** handleOtBookingAlert()	1. An alert with the message "Patient not Selected! Please Select the patient first!" is displayed. 2. Handle and accept the alert to proceed.

Test Case No.	Test Case Name	Test Steps to be performed	Path & Method Used	Expected Result
3	Verify Patient Overview Page Displays Information Correctly	1. Use verifyPatientOverview(). 2. Read data from PatientName.json file for any one patient name. 3. Goto “Doctor” module, then “In Patient Department” tab. 4. In the search bar, enter the patient name read from patientName.json file and perform the search. 5. Locate the patient in the results and click on the “Preview” icon under the Actions column.	**Reference path** \PageObject\Pages \DoctorPage **methods** verifyPatientOverview()	Verify the same patient overview page is displayed with the same patient’s name.
4	Add Progress Note for In Patient	1.Use addProgressNoteForPatient() to check pop up message. 2. Go to “Doctor” module, then “In Patient Department” tab. 3. In the search bar, enter the patient’s name read from patientName.json file and perform the search. 4. Locate the patient in the results and click on the “Preview” icon under the Actions column. 5. Click on “Notes” section. 6. Click on “Add Notes” button. 7. Select “Progress Note” option from “Template” dropdown. 8. Enter subjective Notes as "Test Notes" and click on save button.	**Reference path** \PageObject\Pages \DoctorPage **methods** addProgressNoteForPatient()	The method should successfully add a Progress Note for the patient, and a success confirmation message with the text "Progress Note Template added." should be displayed.
5	Add and Verify New Currency in Settings	1. Navigate to Procurement > Settings. 2. Select "Currency" sub tab. 3. Click "Add Currency" button. 4. Add any data in “Currency Code” and “Description” fields. 5. Click on “Add Currency” button.	**Reference path** \PageObject\Pages\ProcurementPage **methods** addCurrencyAndVerify()	The new currency should be added successfully and displayed in the table with the correct currency code and description.
6	Verify Warning Popup for Mandatory Fields in Scheme Refund	1. Navigate to Utilities module and select "Scheme Refund" tab. 2. If required, please select any counter value and then select “Scheme Refund” tab. 3. Click on "New scheme Refund Entry" button. 4. Now click on save without entering value in any field.	**Reference path** \PageObject\Pages\UtilitiesPage **methods** verifyMandatoryFieldsWarning()	A warning popup should appear with the message: "Please fill all the mandatory fields."
7	Verify Navigation to User Profile Page	1. Navigate to Homepage i.e https://healthapp.yaksha.com/Home/Index#/ 2. Click on the Admin dropdown. 3. Select the "My Profile" option.	**Reference path** \PageObject\Pages\AdminPage **methods** verifyUserProfileNavigation()	Verify that the user is redirected to the "User Profile" page and the page header or title confirms this.
8	Verify Patient Profile Picture Upload	1. Navigate to Patient module. 2. Select “Register Patient” tab. 3. Select “Profile Picture” tab (camera icon).	**Reference path** \PageObject\Pages\PatientPage	Verify that the uploaded image is displayed successfully in the patient's profile.

Test Case No.	Test Case Name	Test Steps to be performed	Path & Method Used	Expected Result
		4. Click on the "New Photo" button. 5. Upload an image present in TestImage folder and click on the "Done" button.	**methods** uploadProfilePicture()	
9	Verify TDS Percent update for an employee	1. Navigate to the “Incentive” module and then “Settings” tab. 2. Click on the "Settings" tab. 3. Locate the row corresponding to the specified employee name. 4. Click the "Edit TDS%" button within the located row. 5. In the "Edit TDS Percent" modal, enter the updated TDS% value. 6. Click on the "Update TDS" button. 7. Verify that the updated TDS% value is correctly displayed in the table.	**Reference path** \PageObject\Pages\IncentivePage **methods** editTDSForEmployee()	The updated TDS% value is displayed correctly in the corresponding row of the table.
10	Verify Price Category Enable/Disable	1. Navigate to “Settings” module. 2. Click on more... and select "Price Category" tab. 3. Click on “Disable” button to disable any Code in the table. 4. Verify a success message appears with the message "Deactivated.". 5. Activate the same code by clicking “Activate” button and verify the success message as “Activated”.	**Reference path** \PageObject\Pages\SettingsPage **methods** togglePriceCategoryStatus()	A success message is displayed for both actions: "Deactivated." for disabling and "Activated." for enabling.
11	Verify Navigation Between Different Tabs	1. Navigate to “Substore” module. 2. Click on “Accounts” button. 3. Select “Inventory” tab. 4. Click and navigate between different tabs like “Stock”, “Inventory Requisition”, “Consumption”, “Reports”, “Patient Consumption” and “Return”.	**Reference path** \PageObject\Pages\SubstorePage **methods** verifyNavigationBetweenSubmodules()	Ensure that it should navigate to each sub tab of the "Inventory" tab.
12	Verify tooltip text on hover in Inventory tab.	1. Navigate to “Substore” module. 2. Click on “Accounts” button. 3. Hover over a black coloured icon on top right side and capture the text.	**Reference path** \PageObject\Pages\SubstorePage **methods** verifyTooltipText()	Tooltip text should contain: "You are currently in Accounts sub store. To change, you can always click here."
13	TS-13 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** \PageObject\Pages\SubstorePage **methods** captureInventoryRequisitionScreenshot()	Screenshot of the page is captured and saved successfully.
14	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 form PatientName.json file. 4. Click on “...” button from table and select “Change Doctor”.	**Reference path** \PageObject\Pages\ADTPage **methods** verifyFieldLevelErrorMessage()	Verify a field level error message appears "Select doctor from the list."

Test Case No.	Test Case Name	Test Steps to be performed	Path & Method Used	Expected Result
		5. Change doctor modal will open and then click on update button without filling any value.		

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

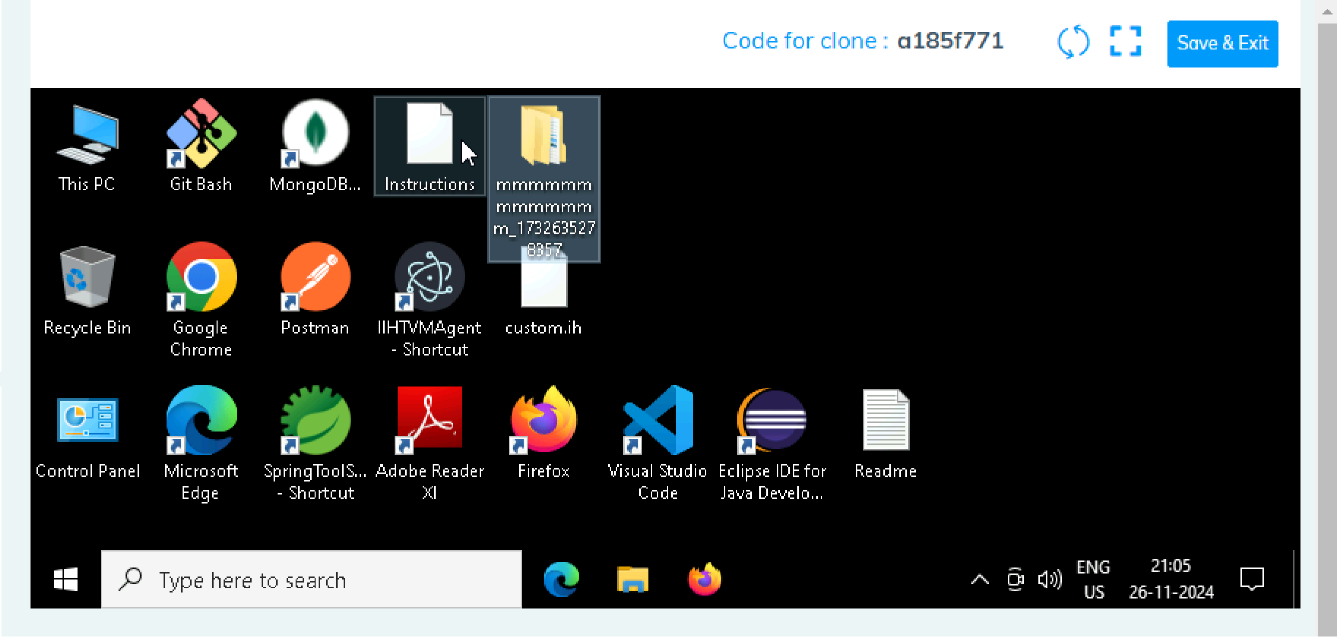
1. Associates should have written clean code that is readable.
2. Associates need to follow SOLID programming principles.

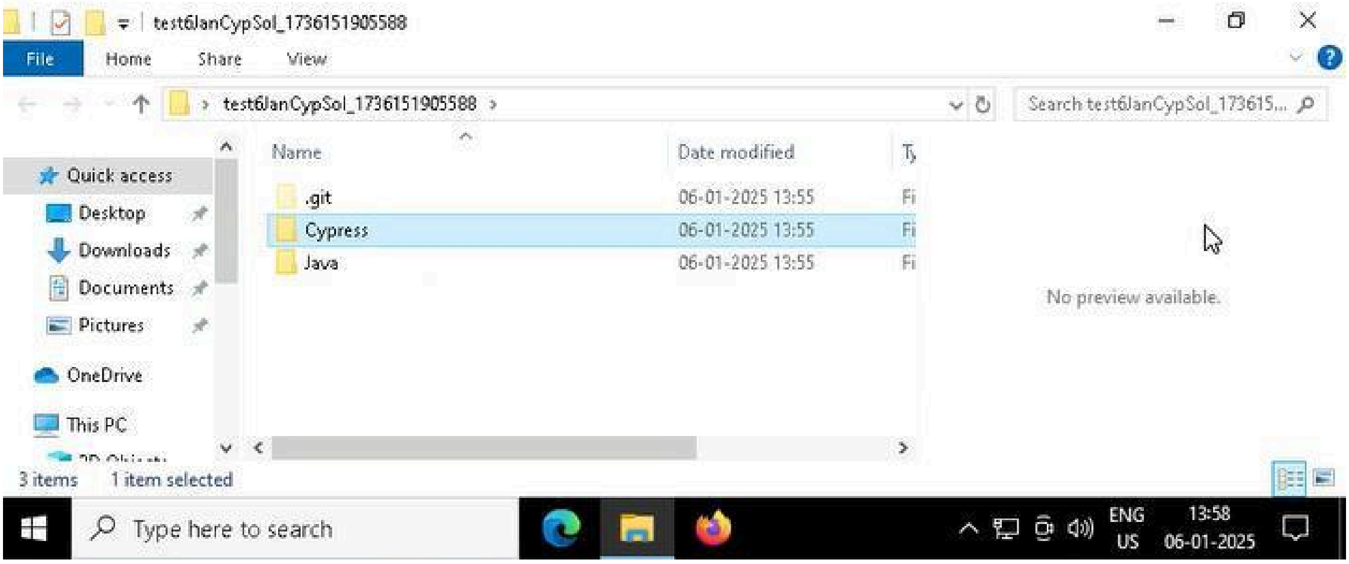
Execution Steps:

Mymedic automation using cypress

Steps for Execution:

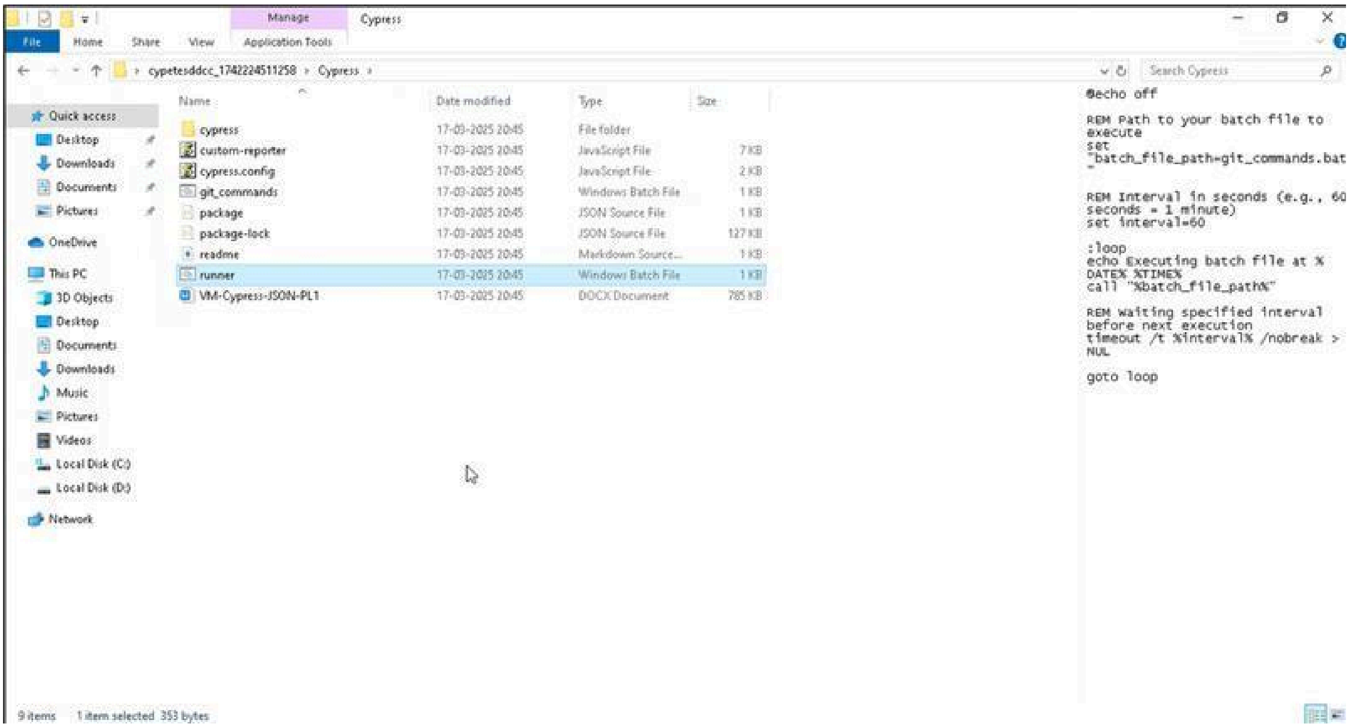
1. **Please open the folder created on desktop with the email name you used to login.**





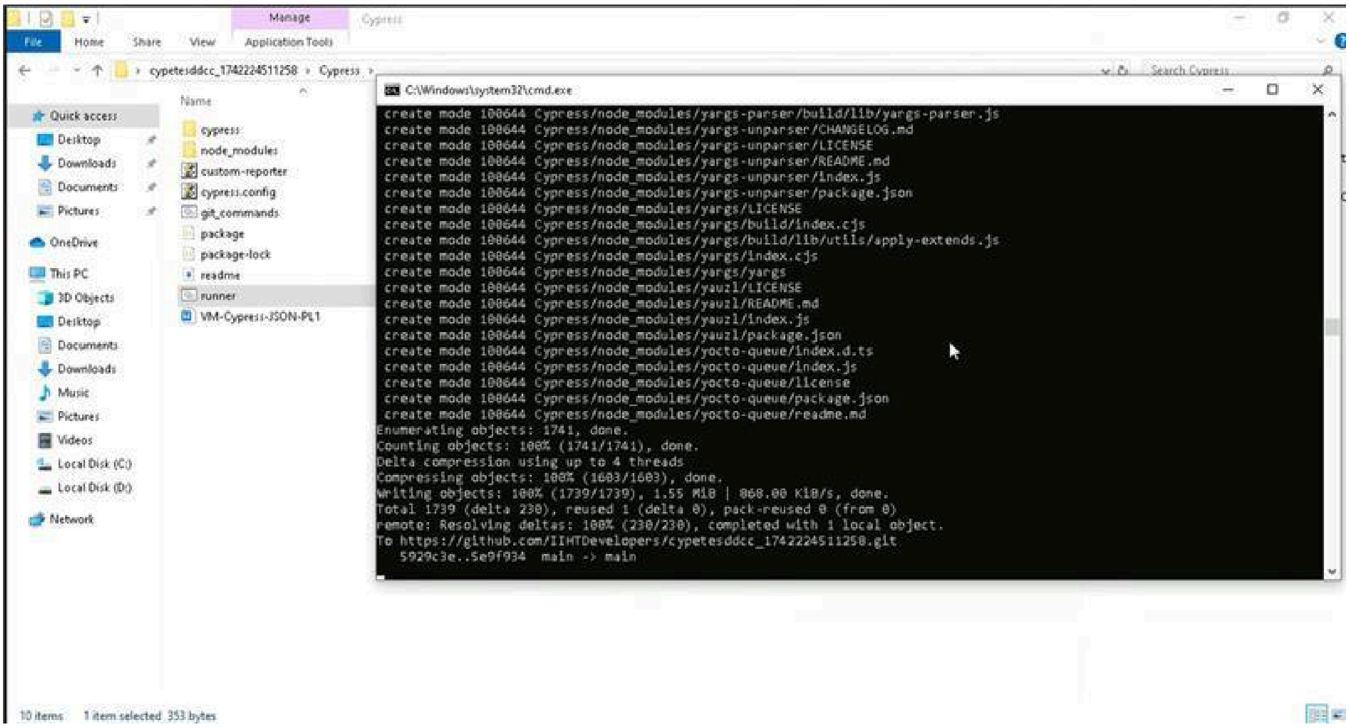
2. **Go into the Cypress folder and execute this “runner” file. This will keep pushing the code at regular**

intervals.



Name	Date modified	Type	Size
cypress	17-03-2025 20:45	File folder	
custom-reporter	17-03-2025 20:45	JavaScript File	7 KB
cypress.config	17-03-2025 20:45	JavaScript File	2 KB
git_commands	17-03-2025 20:45	Windows Batch File	1 KB
package	17-03-2025 20:45	JSON Source File	1 KB
package-lock	17-03-2025 20:45	JSON Source File	127 KB
readme	17-03-2025 20:45	Markdown Source...	1 KB
runner	17-03-2025 20:45	Windows Batch File	1 KB
VM-Cypress-JSON-PL1	17-03-2025 20:45	DOCX Document	785 KB

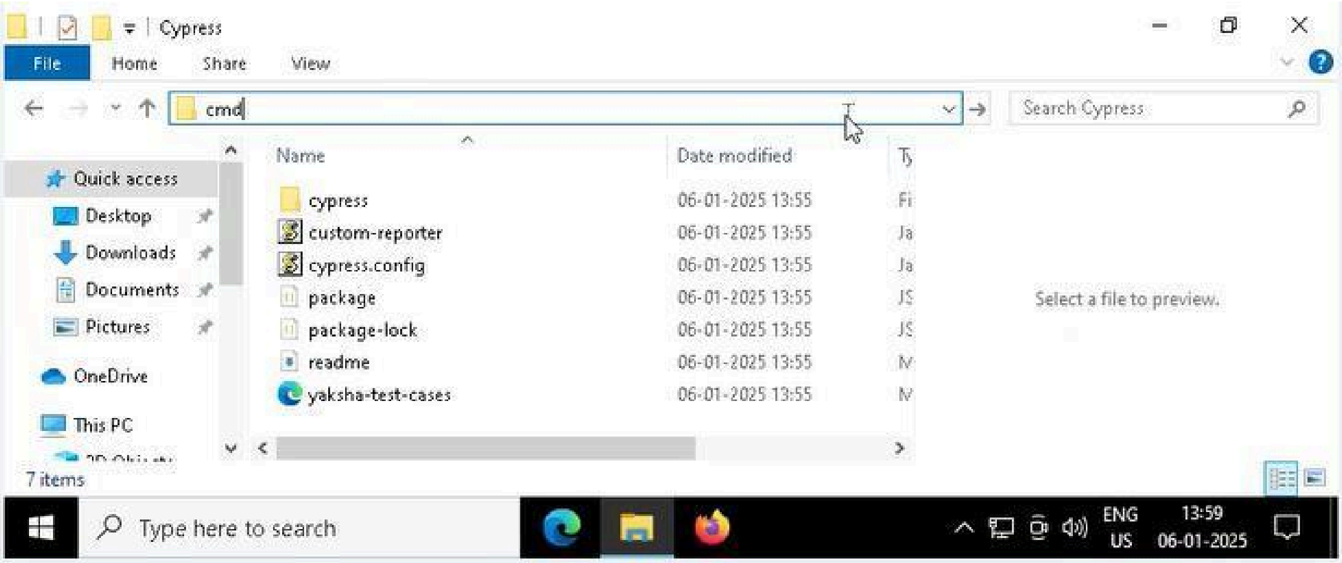
```
@echo off

REM Path to your batch file to
execute
set
"batch_file_path=git_commands.bat
"

REM Interval in seconds (e.g., 60
seconds = 1 minute)
set interval=60

:loop
echo Executing batch file at %
DATE% %TIME%
call "%batch_file_path%"

REM waiting specified interval
before next execution
timeout /t %interval% /nobreak >
NUL

goto loop
```



```
 create mode 100644 Cypress/node_modules/yargs-parser/build/lib/yargs-parser.js
 create mode 100644 Cypress/node_modules/yargs-unparser/CHANGELOG.md
 create mode 100644 Cypress/node_modules/yargs-unparser/LICENSE
 create mode 100644 Cypress/node_modules/yargs-unparser/README.md
 create mode 100644 Cypress/node_modules/yargs-unparser/index.js
 create mode 100644 Cypress/node_modules/yargs-unparser/package.json
 create mode 100644 Cypress/node_modules/yargs/LICENSE
 create mode 100644 Cypress/node_modules/yargs/build/index.cjs
 create mode 100644 Cypress/node_modules/yargs/build/lib/utils/apply-extends.js
 create mode 100644 Cypress/node_modules/yargs/index.cjs
 create mode 100644 Cypress/node_modules/yargs/yargs
 create mode 100644 Cypress/node_modules/yauzl/LICENSE
 create mode 100644 Cypress/node_modules/yauzl/README.md
 create mode 100644 Cypress/node_modules/yauzl/index.js
 create mode 100644 Cypress/node_modules/yauzl/package.json
 create mode 100644 Cypress/node_modules/yocto-queue/index.d.ts
 create mode 100644 Cypress/node_modules/yocto-queue/index.js
 create mode 100644 Cypress/node_modules/yocto-queue/license
 create mode 100644 Cypress/node_modules/yocto-queue/package.json
 create mode 100644 Cypress/node_modules/yocto-queue/readme.md
Enumerating objects: 1741, done.
Counting objects: 100% (1741/1741), done.
Delta compression using up to 4 threads
Compressing objects: 100% (1603/1603), done.
Writing objects: 100% (1739/1739), 1.55 MiB | 868.00 KiB/s, done.
Total 1739 (delta 230), reused 1 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (230/230), completed with 1 local object.
To https://github.com/IIHTDevelopers/cypetesddcc_1742224511258.git
   5929c3e..5e9f934  main -> main
```

3. **Open command prompt with its location and use below command:**

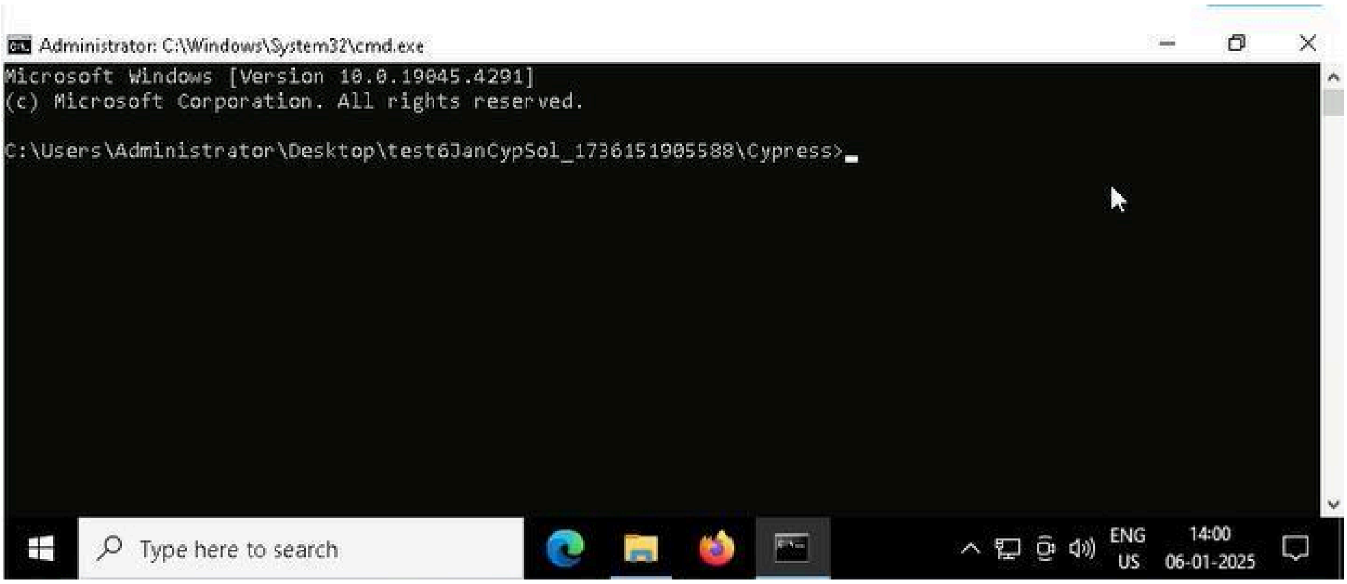
   code .



Cypress

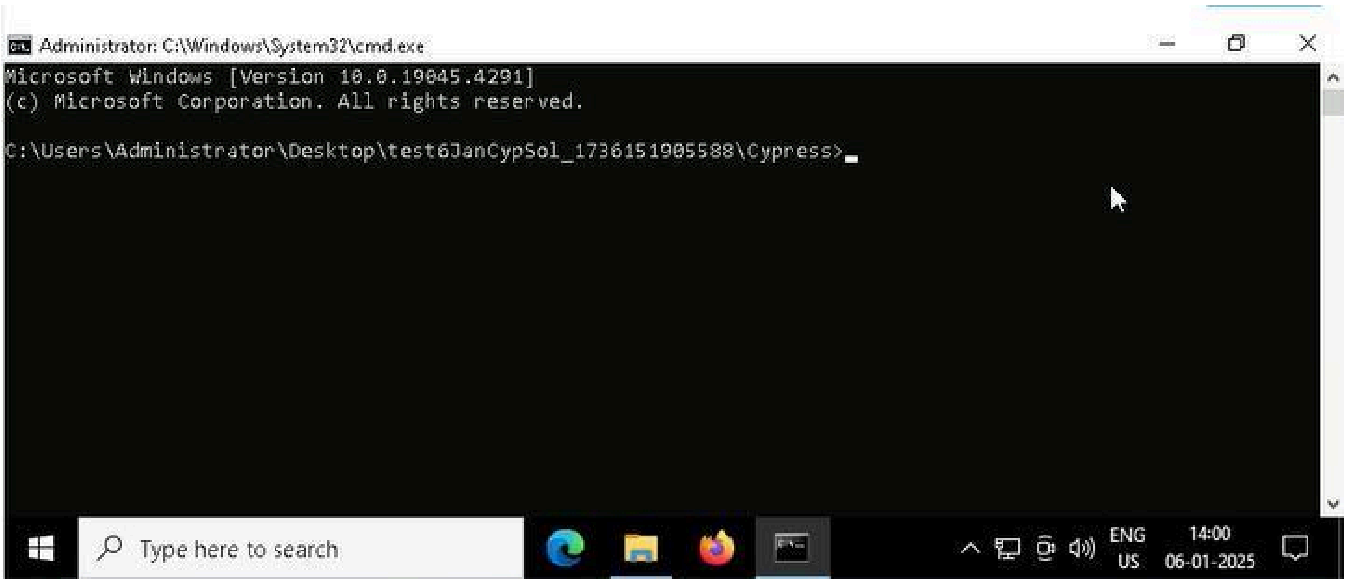
cmd

Name	Date modified	Ty
cypress	06-01-2025 13:55	Fi
custom-reporter	06-01-2025 13:55	Ja
cypress.config	06-01-2025 13:55	Ja
package	06-01-2025 13:55	JS
package-lock	06-01-2025 13:55	JS
readme	06-01-2025 13:55	M
yaksha-test-cases	06-01-2025 13:55	M



Administrator: C:\Windows\System32\cmd.exe

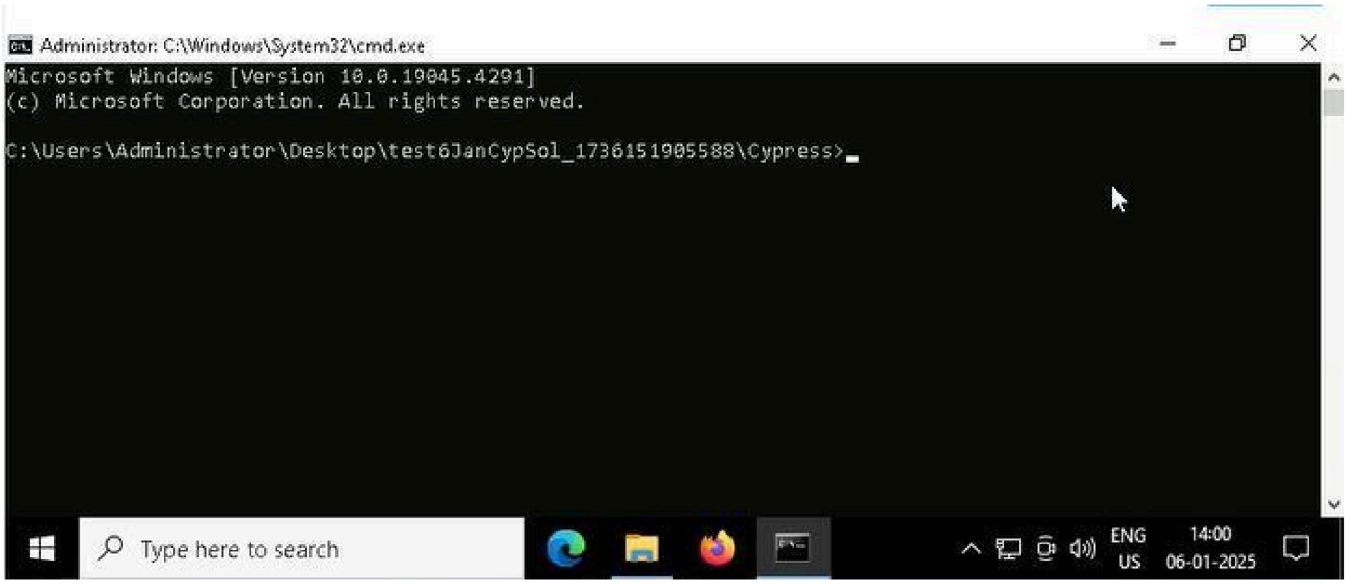
```
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

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



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

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

Mymedic automation using cypress

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

5. 

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

C:\Users\Administrator\Desktop\test6JanCypSol_1736151905588\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:**

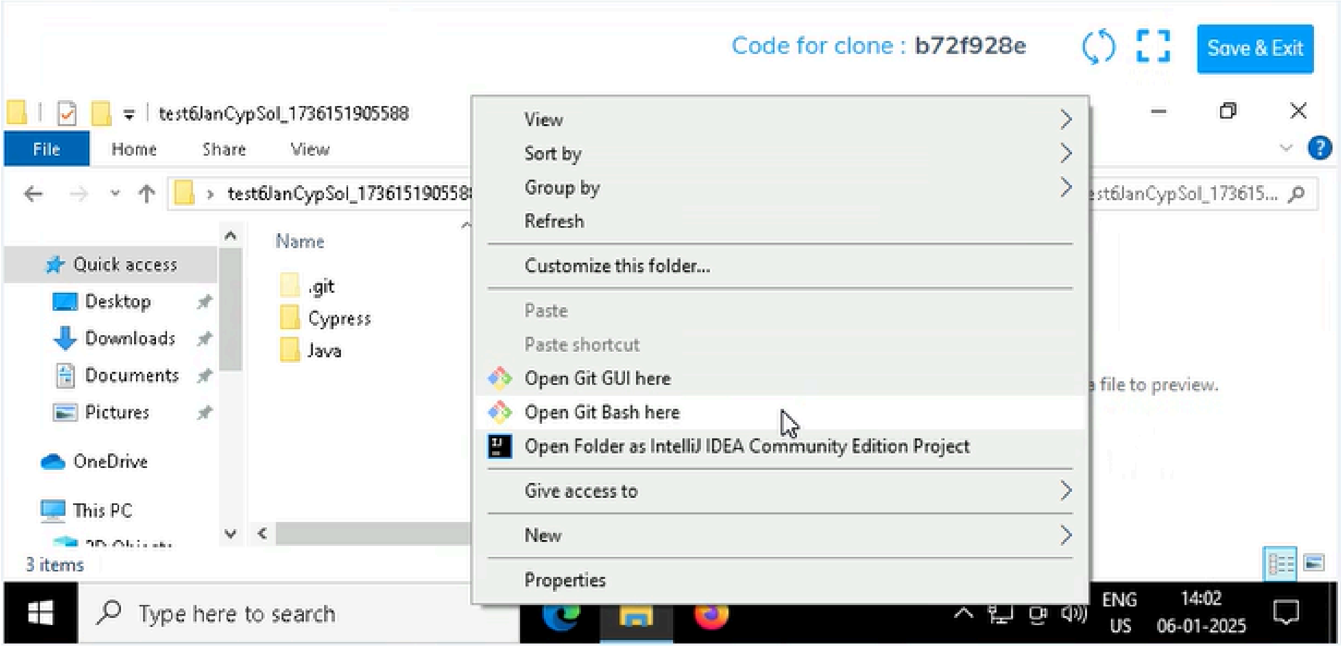
   npx cypress open

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

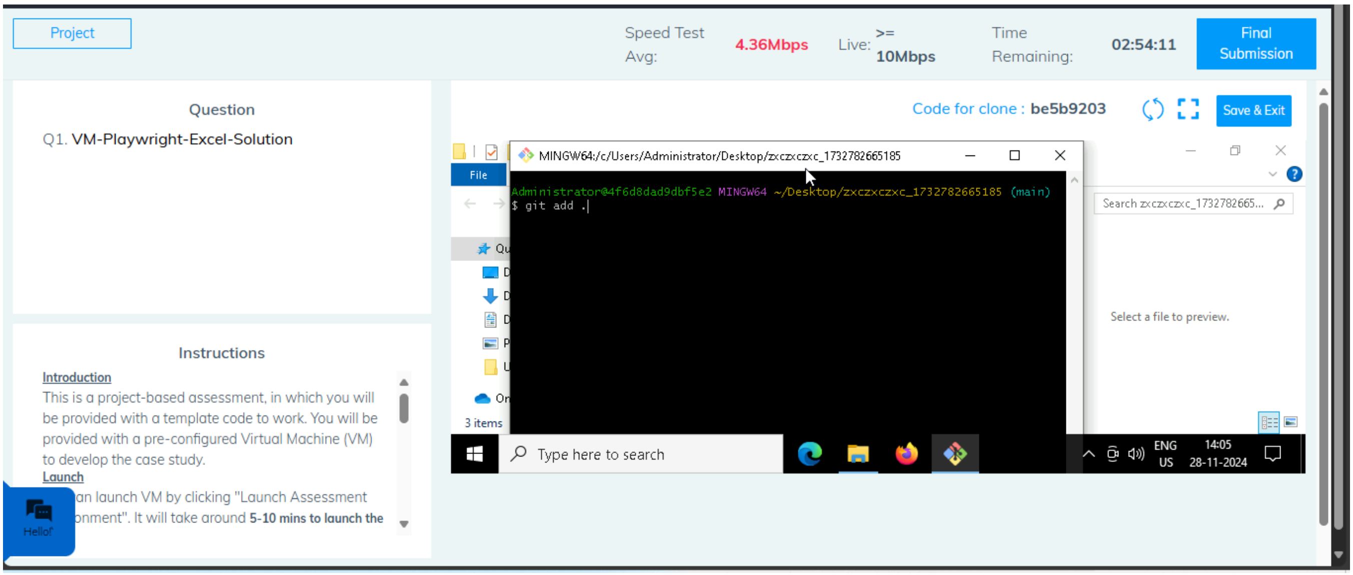
   npx cypress run

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**



2. **Add all files**



```
Administrator@4f6d8dad9dbf5e2 MINGW64 ~/Desktop/zxczxczxc_1732782665185 (main)
$ git add .
```

Question

Q1. VM-Playwright-Excel-Solution

Instructions

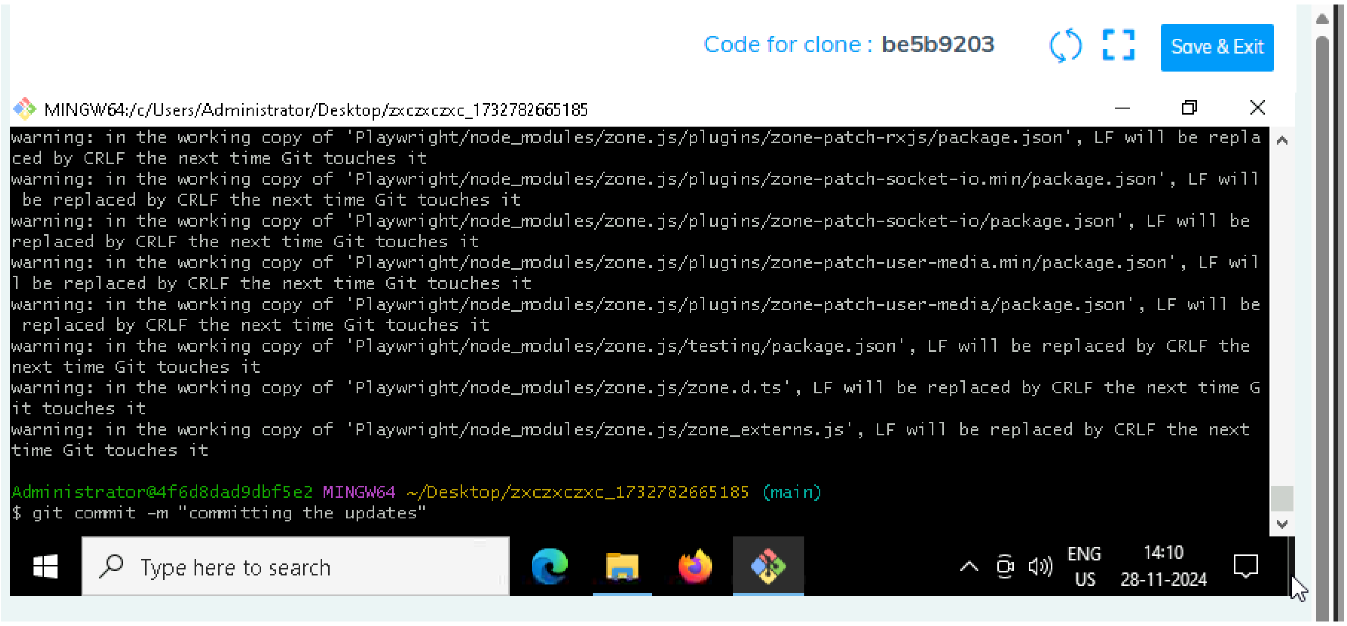
Introduction

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

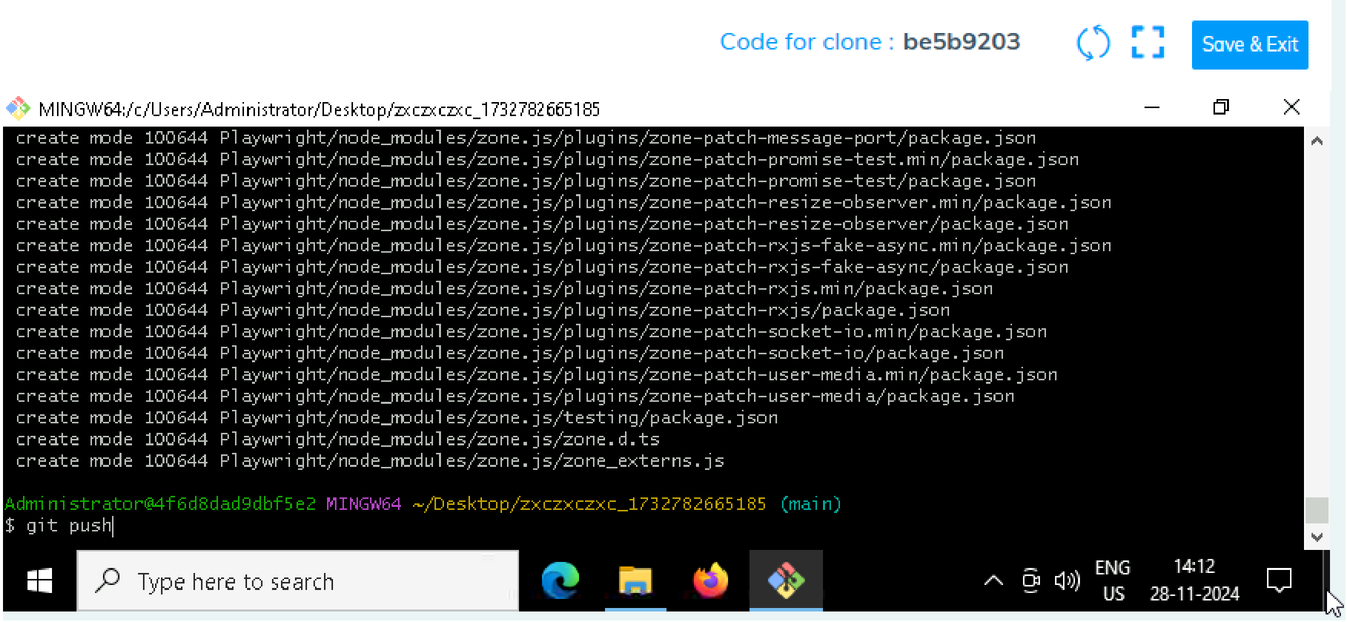
an launch VM by clicking "Launch Assessment onment". It will take around **5-10 mins to launch the**

3. **Commit the changes**


Code for clone : be5b9203
Save & Exit
MINGW64:/c/Users/Administrator/Desktop/zxczxczxc_1732782665185
Type here to search
ENG
US
14:10
28-11-2024

```
warning: in the working copy of 'Playwright/node_modules/zone.js/plugins/zone-patch-rxjs/package.json', LF will be repla
ced 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 wil
l 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 G
it 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/zxczxczxc_1732782665185 (main)
$ git commit -m "committing the updates"
```

4. **Push the changes**


Code for clone : be5b9203
Save & Exit
MINGW64:/c/Users/Administrator/Desktop/zxczxczxc_1732782665185
Type here to search
ENG
US
14:12
28-11-2024

```
 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/zxczxczxc_1732782665185 (main)
$ git push
```

Mymedic automation using cypress