

YAKSHA HEALTH APP WITH PYTHON AND PYTEST

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:** Playwright (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:

1. **Login/Registration:** For user authentication.
2. **Dashboard:** Main page displaying patient records.
3. **Patient Records:** Add, edit, delete, and view records.
4. **Appointment Scheduling:** Manage patient appointments.
5. **Search/Filter:** Filter records by various parameters (e.g., doctor, date).

Project Information:

- **Use Case:** Aimed at simplifying EMR management, enhancing healthcare record accessibility, and enabling easy interaction with patient data for healthcare providers

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

Here's a detailed table format for the test cases to be tested

We have placed an json file on desktop along with this file containing few fields which should be used while implementing.

S No.	Summary	Action	Expected Result
1	Verify the user can search patient under appointment module and the searched patient is filtered out.	<ol style="list-style-type: none"> 1. Click on the appointment link. 2. Wait for the appointment page to load. 3. Identify the list of counter items and select the first one (if available). 4. Retrieve the patient's name from the displayed appointment details. 5. Enter the retrieved patient name into the search bar and submit the search. 6. Wait for the search results to load. 7. Extract the patient name from the search result and compare it with the original name. 	<ol style="list-style-type: none"> 1. The search result should display the exact patient name that was entered in the search bar. 2. The function should return True if the names match and False otherwise. 3. No exceptions should occur during execution.
2	Verify active counter message under Dispensar moduel	<ol style="list-style-type: none"> 1. Click on the Dispensary module link. 2. Click on the "Activate Counter" button. 3. Wait for the page to load and ensure the counters are visible. 4. Count the available counters. 5. If counters are available: <ol style="list-style-type: none"> 5.1. Select a random counter. 5.2. Click to activate the selected counter. 5.3. Verify that the activation message contains the selected counter name. 	<ol style="list-style-type: none"> 1. A counter should be successfully activated. 2. The activation message should display the correct counter name. 3. No errors should occur during the process.
3	Verify the components of purchase requiest list are visible correctly under procurement module	<ol style="list-style-type: none"> 1. Navigate to the Procurement module. 2. Verify the visibility of the following elements: <ol style="list-style-type: none"> 2.1. Purchase Request 2.2. Purchase Order 2.3 Goods Arrival Notification 2.4 Quotations 2.5 Settings 2.6 Reports 2.7 Favorite Button 2.8 OK Button 2.9 Print Button 2.10. First Button 2.11. Previous Button 2.12. Next Button 2.13. Last Button 3. Select "01-01-2020" in the From Date field. 4. Click the OK button. 5. Verify if all elements are visible on the page. 	<ol style="list-style-type: none"> 1. All required elements should be present and visible. 2. Entering the date and clicking OK should not cause any errors. 3. The verification process should complete without missing elements.
4	Verify the user gets error one adding new lab test without filling details	<ol style="list-style-type: none"> 1. Navigate to Laboratory > Settings. 2. Click on "Add New Lab Test". 3. Click on the "Add" button without entering any values. 4. Capture the error message displayed. 5. Verify that the error message is visible and correctly logged. 	<ol style="list-style-type: none"> 1. An error message should be displayed indicating that required fields are missing. 2. The "Add Lab Test" modal should remain open. 3. The error message should be clearly visible and correctly captured.

5	Perform a radiology request and handle alert	<ol style="list-style-type: none"> 1. Navigate to the Radiology Module. 2. Click on List Request sub-module. 3. Enter a From Date and apply the filter by clicking the OK button. 4. Click on the "Add Report" button in the list (if Available). 5. Close the Add Report Modal using the Close button. 6. If the Close button remains visible, click it again. 7. Press Shift + Tab three times to navigate backward. 8. Press Enter to confirm or close. 9. Handle any alert messages that appear: <ol style="list-style-type: none"> 9.1. If the alert message says "Changes will be discarded. Do you want to close anyway?", accept the alert. 9.2. Otherwise, dismiss the alert. 10. Repeat steps 1-9 for a second iteration to ensure the process handles multiple requests correctly. 	<ol style="list-style-type: none"> 1. The radiology request should be performed successfully without errors. 2. If an alert appears, it should be handled based on its message content. 3. The modal should close properly when the close button is clicked. 4. The process should run twice without failure, ensuring repeatability.
6	Verify the user can search and verify the patients exists on the patient module	<ol style="list-style-type: none"> 1. Navigate to the Patient Section. 2. Locate the search bar. 3. Iterate over a predefined list of patient names and for each patient: <ol style="list-style-type: none"> 3.1. Enter the patient name in the search bar. 3.2. Press Enter to initiate the search. 3.3. Capture the search result displayed in the patient grid. 3.4. Compare the result with the expected patient name. 3.5. If the name matches, proceed to the next patient; otherwise, log the mismatch and fail the test. 3.6 Clear the search bar before searching for the next patient. 	<ol style="list-style-type: none"> 1. The search results should display the exact patient name that was searched for. 2. All patient searches should be executed successfully without errors. 3. If a mismatch occurs, the test should fail and log the discrepancy.
7	Verify notice message after entering incorrect filters	<ol style="list-style-type: none"> 1. Navigate to the Procurement Module. 2. Click on the Purchase Request tab. 3. Enter an invalid date ("00-00-0000") in the From Date field. 4. Click the OK button to apply the filter. 5. Capture and validate the error message displayed. 	<ol style="list-style-type: none"> 1. The system should display an error message stating: "Date is not between Range. Please enter again" 2. The invalid filter should not be applied. 3. The user should be prompted to correct the date input.
8	Verify Maternity Allowance Payment Receipt Modal	<ol style="list-style-type: none"> 1. Navigate to the Maternity Module. 2. Click on the Reports submodule. 3. Select the Maternity Allowance Report. 4. Enter the From Date (e.g., 01-01-2020). 5. Click the Show Report button to generate the report. 6. Click View Details for the first record. 	<ol style="list-style-type: none"> 1. The Maternity Allowance Report should load successfully. 2. Clicking View Details should open the Maternity Allowance Payment Receipt modal. 3. The modal should be visible and display the correct information.
9	Verify Error Message When Editing an Occupied Bed	<ol style="list-style-type: none"> 1. Navigate to the Settings Module. 2. Click on the ADT (Admission, Discharge, and Transfer) Link. 3. Click on the Manage Bed tab. 4. Attempt to edit an occupied bed by clicking the Edit button. 	<ol style="list-style-type: none"> 1. An error message should be displayed stating: "Cannot modify occupied beds." 2. The bed details should not be editable. 3. No unexpected errors should occur.

10	Verify Add New Patient modal opens with Alt + Enter keyboard shortcut	1. Navigate to the Dispensary Module. 2. Click on the Activate Counter page. 3. Wait for the page to load. 4. Press Alt + N to select the Sale tab.	1. The Sale tab should be selected. 2. The Patient modal should appear successfully. 3. No errors should occur during the process.
11	Verify Remarks text field placeholder value	1. Navigate to the Operation Theatre Module. 2. Click on the New OT Booking button. 3. Check if the Remarks text area is visible and enabled. 4. Verify that the Remarks text area has the correct placeholder text ("Remarks").	1. The New OT Booking page should load successfully. 2. The Remarks text area should be enabled for user input. 3. The placeholder text in the Remarks text area should be "Remarks".
12	Verify Daily Transactions Report for	1. Navigate to the Accounting Module. 2. Click on the Reports link. 3. Click on the Daily Transactions Report. 4. Select the fiscal year (e.g., 2023) from the dropdown. 5. Click the Load button.	1. The Daily Transactions Report should load successfully. 2. The results table should be visible. 3. No errors should occur during the loading process.
13	Verify Navigation to Patient Overview from Past Days Records	1. Navigate to the Nursing Module. 2. Click on the Past Days tab. 3. Enter the From Date (e.g., 01-01-2020). 4. Click the OK button. 5. Enter "Deepika Rani" in the search input field. 6. Wait for search results to load. 7. Click on the patient's record. 8. Click on Overview from the Actions column.	1. The system should display the Overview Page for the patient "Deepika Rani". 2. The patient overview section should be visible without errors. 3. No unrelated patients should appear in the search results.
14	Verify table filtering for "Male Ward"	1. Navigate to the Lab Dashboard. 2. Click on the Sample Collections tab. 3. Enter the From Date (e.g., 01-01-2020). 4. Click the OK button. 5. Hover over the Item column. 6. Click the Hamburger Menu to open filter options. 7. Select Starts with or Equals from the dropdown filter. 8. Enter "Male Ward" in the text field.	1. The sample collections table should display only the records that match the From Date and Item filter criteria (e.g., "Male Ward"). 2. No unrelated records should appear. 3. The filtering operation should complete without errors.

MAKE SURE TO HAVE THE JSON FILE AVAILABLE IN THE VM DESKTOP WITHIN THE CREATED PROJECT FOLDER

EXPECTATIONS:

Learners will gain experience in building strongly-typed applications using Python, Selenium, and managing their dataflow. They'll learn how to define interfaces, use types for error prevention, and improve code maintainability. app. Key skills include:

- **Browser Automation:** Interacting with web elements and testing multiple browsers.
- **Assertions & Validations:** Ensuring app behavior meets expected results.

- **End-to-End Testing:** Automating real user interactions and validating overall app functionality.

STEPS TO FOLLOW

- After you login from in to the virtual machine, you need to open the created folder and open the command prompt in that directory
- Using the command code . open VS code
- Write the code in the template as the requirements given above and run the code
- Open the Terminal
- To run the code, python filename.py
- Execute the following command to run all test cases:
- pytest

2. Run all test cases:

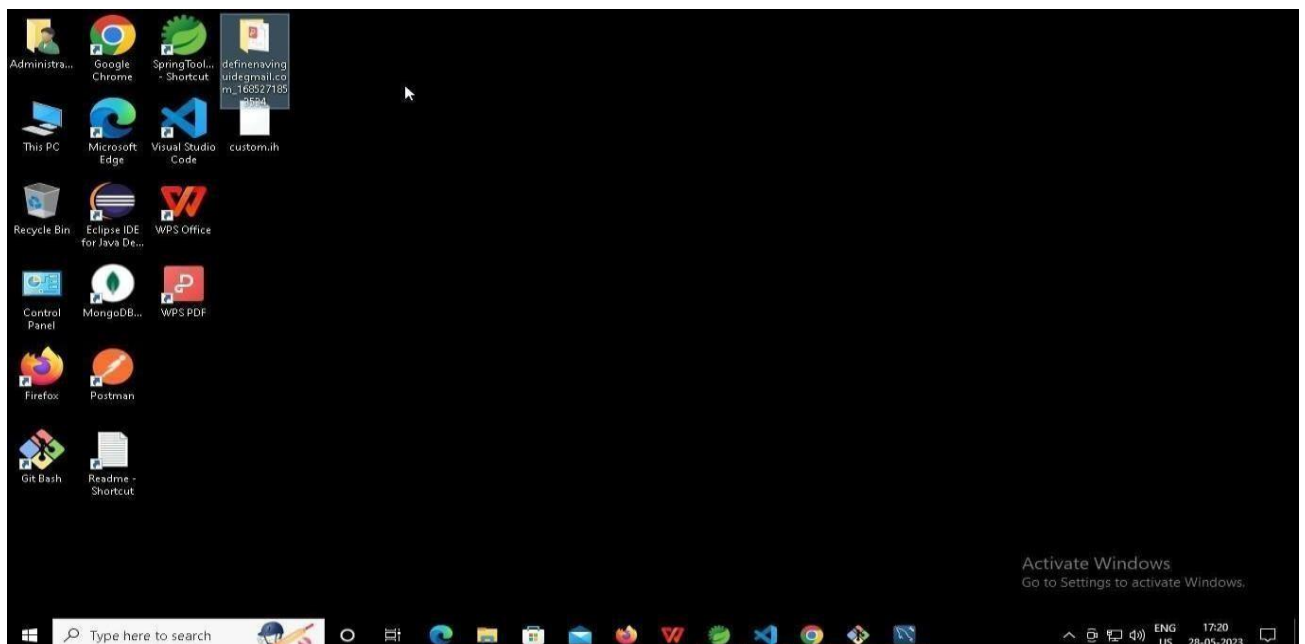
Using the command `pytest`

Screenshot to push the application to github

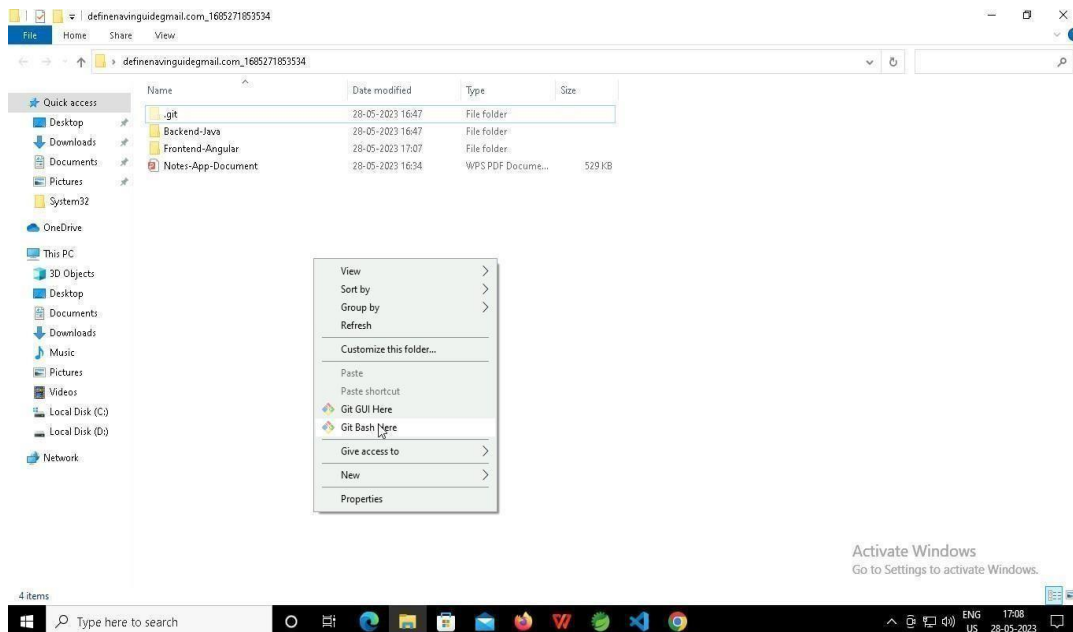
X _____

You can run test cases as many numbers of times and at any stage of Development, to check howmany test cases arepassed/failed and accordingly refactor your code.

1. Make sure before final submission you commit all changes to git. For that open theproject folder available on desktop



- Right click in folder and open Git Bash



b. In Git bash terminal, run following commands

c. `git status`

```

Administrator@2a5ee7ad258f58c MINGW64 ~/Desktop/tesgmailcom_1728449791471 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    templateespark.py

no changes added to commit (use "git add" and/or "git commit -a")
Administrator@2a5ee7ad258f58c MINGW64 ~/Desktop/tesgmailcom_1728449791471 (main)
$

```

d. `git add .`

```

Administrator@2a5ee7ad258f58c MINGW64 ~/Desktop/tesgmailcom_1728449791471 (main)
$ git add .

```

e. `git commit -m "First commit"`
(You can provide any message every time you commit)

```
Administrator@2a5ee7ad258f58c MINGW64 ~/Desktop/tesgmailcom_1728449791471 (main)
$ git commit -m "first commit"
[main f97ce24] first commit
1 file changed, 91 deletions(-)
delete mode 100644 templateespark.py
```

f. git push

```
Administrator@2a5ee7ad258f58c MINGW64 ~/Desktop/tesgmailcom_1728449791471 (main)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 212 bytes | 212.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/IIHTDevelopers/tesgmailcom_1728449791471.git
a1c1905..f97ce24 main -> main
```

After you have pushed your code Finally click on the final submission button
