1. Usability and Functionality Testing:

- Login Interface:
  - □ Provided credentials: Username: tester; Password: tester2023!
  - □ Opinion:
    1. The login interface is simple and visually appealing.
    2. Fields are easy to identify and perfectly labeled.
    3. Doesn't need many information/ credentials for logging in.
    4. The user can easily recover his/her password if s/he forgets the password. One needs to click "Forgot password" button and then there appears another interface where one needs to provide the username and cellphone number, including the code. There is dropdown menu for selecting codes where user can choose any of two options between "ZM" and "AO".
    5. Moreover, there is a "Remember me" button, which actually saves the login details of the user. It is crucial for the user to login easily, and it saves time, which is a plus.
    6. Additionally, if the user is new, h/she can create a new account by clicking the "Sign up" button below.
    7. Lastly, there is a theme button in the top right corner where the user can change themes among three different options: light, dark, and gray.
    8. All these functions mentioned above work perfectly fine.
  - □ Issues:
    1. The "Help Desk" function below doesn't respond.

- Select Facility Interface:
  - □ Provided credentials: Province: Lusaka; District: Lusaka, Facility: Dr. Watson Dental Clinic
  - □ Issues:
    1. The facility field search bar box's suggestions are not keyboard-friendly. These can only be chosen by mouse hovering; the keyboard's up and down arrow keys do not work for this.

- User Profile Registration:
  - □ Observations:
    1. Form fields are clearly labeled.
    2. Dropdown options in the "Sex" field sped up data entry.
    3. If a primary field is missed and the user tries to continue, then an error message is displayed on the screen, which

makes the user understand their mistakes in filling out the form.
4. Realtime validation in "Cellphone" field is effective.

□ No issues were observed.
□ Recommendation: The color contrast for gray theme can be improved.

- Search or add new patient:
  □ Observations:
  1. There is an option for biometrics where the user can select a finger and click on the icon to scan the fingerprint. This is crucial for ensuring the identification of patients in any hospital or healthcare system. The instruction message was clear.
  2. A patient can be searched through NRC, NUPN, Cellphone or Full Name. If a patient is not found, then the add new patient option appears, which takes the user to the User Profile Registration.
  3. Searched information was displayed in an organised manner.
  4. The client can edit the patient's information easily by clicking on the edit button. In the admission section, the patient's past and current admissions to various departments and wards are provided.

  □ Issues:
  1. In the "admission" section, there is a dropdown menu for "Department" and "Ward," where only few options are available. But if we see the details below, there are other options than the ones in the dropdown menu. So these need to be included in the dropdown menu.

  □ Recommendations:
  1. A "FAQ" section or "Help Desk" would be helpful for new users.

- **NRC = 111111/11/1 > ATTEND TO PATIENT > VITAL:**

  □ Issues:
  1. The UI is not easy to understand or user-friendly.
  2. Vital>Details>Edit: BP unrecordable field is not editable.

  □ Recommendations:
  1. Users would find it easier to read the summary if the "Recent Data Summary" section was presented horizontally as opposed to vertically.

2. It would be great if Vital>Details> Edit: The BP unrecordable field could be edited by fixing it.

2. Automated Testing:

- **Automated Data Upload:**

  This Selenium test script named ‘**automatedataupload.py**’ automates the process of logging into a web application, selecting a province, district, and facility, entering patient details, and uploading a sample medical dataset.

  Prerequisites:
  - Python installed
  - Selenium library (‘pip install selenium’)
  - Chromedriver installed and added to path

  Usage:
  - Please run the script from the terminal: **python automatedataupload.py**

  Script Structure:
  - initialize_driver(): Initialize WebDriver for Chrome.
  - login(driver, username, password): Log in to the web application.
  - select_province(driver, province_name): Select a province from the dropdown.
  - select_district(driver, district_name): Select a district from the dropdown.
  - select_facility(driver, facility_name): Select a facility from the dropdown.
  - click_enter_button(driver): Click the "Enter" button.
  - click_nrc_button(driver): Click the "NRC" button.
  - enter_nrc_value(driver, nrc_value): Enter NRC value.
  - click_attend_to_patient_button(driver): Click "Attend to Patient" button.
  - click_image_element(driver): Click on an image element.
  - vital_button(driver): Click on the "Vital" button.
  - addvital_button(driver): Click on the "Add Vital" button.
  - upload_sample_dataset(driver, dataset_path): Upload a sample medical dataset.

  Output Video:
  - Please go through the following link to see the output of the script.

    AutomateDataUpload.mp4

- **Test User Authentication:**

  This Selenium test suite named ‘**testuserauthentication.py**’ verifies user authentication functionality on a web application. It includes test cases for valid and invalid login scenarios.

  Usage:
  - Please run the script from the terminal: **python ‘testuserauthentication.py**

  Test Cases:

- test_valid_login:
  - This test case verifies successful login with valid credentials.
- test_invalid_username:
  - This test case verifies login failure with an invalid username.
  - It enters an invalid username and a correct password, then checks for the error message.
- test_invalid_password:
  - This test case verifies login failure with an invalid password.
  - It enters a valid username and an invalid password, then checks for the error message.
- test_invalid_usernamepassword:
  - This test case verifies login failure with both an invalid username and password.
  - It enters invalid credentials and checks for the error message.

<u>Output Video:</u>

- Please go through the following link to see the output of the script.
  🗎 UserAuthentication.mp4

- **<u>Data Manipulation:</u>**

This Selenium test suite '**datamanipulation_change_surname.py**' automates the process of editing a patient's profile on the SmartCarePro web application. It includes methods to handle login, dropdown selections, and form submissions, with a primary test case to verify profile data manipulation. It changes the surname of the person from 'Don' to 'Donut'.

<u>Usage:</u>

- Please run the script from the terminal: **python datamanipulation_change_surname.py**

<u>Test Case:</u>

- test_edit_patient_profile:
  - Logs in with valid credentials.
  - Selects the province, district, and facility from dropdown menus.
  - Navigates to the patient's profile, edits the surname, and submits the changes.
  - Asserts that the profile is updated successfully by checking for a success message.

<u>Test Flow:</u>

- **Login**: The test logs in using valid credentials.
- **Select Location**: The test selects the province, district, and facility from dropdown menus.
- **Edit Profile**: The test navigates to the patient's profile, edits the surname field, and submits the form.
- **Verify Update**: The test checks for a success message to verify that the profile was updated successfully.

<u>Output Video:</u>

- Please go through the following link to see the output of the script.
  🗎 DataManipulation.mp4

- **Invalid Input Field Error Handling:**
  This is designed to automate error handling for invalid NRC inputs on the SmartCarePro web application. It also includes methods for logging in, selecting dropdown options. It simply checks whether the NRC input field allows any input with length of less than 9. It also checks for an error message indicating the invalid input format. There was no error message for any invalid input for this field.

  Output Video:
  - Please go through the following link to see the output of the script.
    - 🗎 InvalidInputField_ErrorHandling.mp4

2. File Import with Form Field Testing:

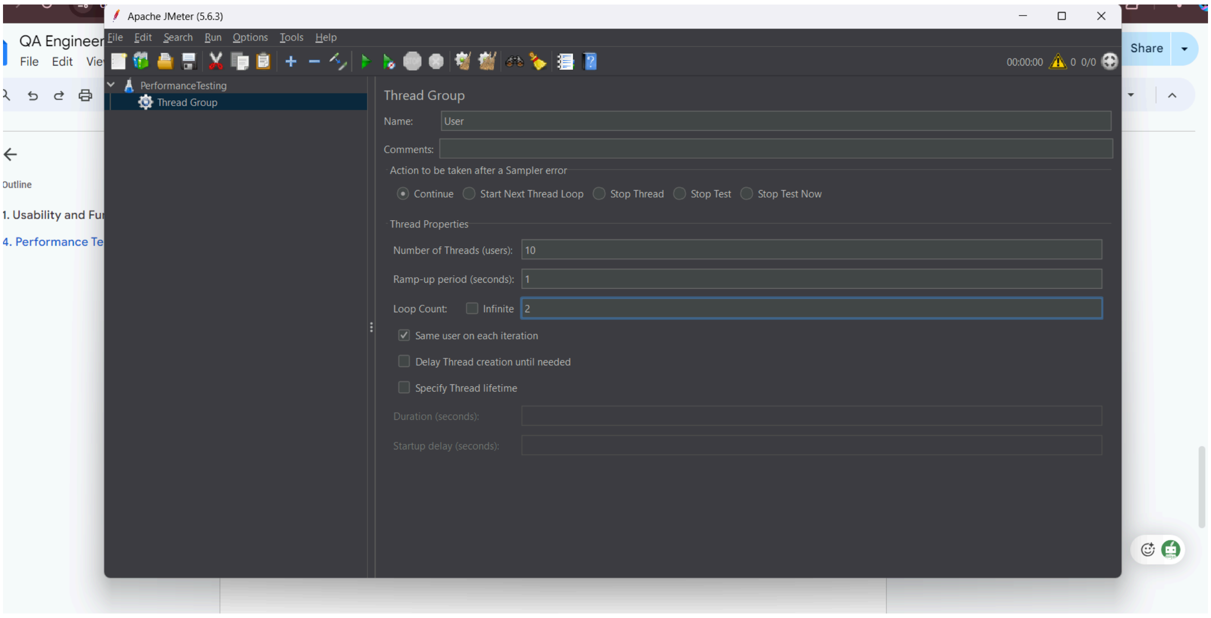
- **Testing the functionality of file import for patient records, ensuring compatibility with various file formats:**
  It focuses on testing the functionality of file import for patient records and validating form fields. Here, I tested it with a '.xlxs' file, and the test result is that the web app doesn't support a '.xlxs' file for filling out the forms. So the error message "Error adding vital" was displayed. The '.xlxs' file is attached to the github file named '**testfileformat.xlxs**'. However, the web app is perfect for the '.csv' file format, which is already tested. Otherwise, the data could not be uploaded from the '**Sample Dataset.csv**' file.

  Output Video:
  - Please go through the following link to see the output of the script.
    - 🗎 Import_XLXS_File.mp4

4. Performance Testing:

I tried to do the load testing by using Jmeter. But I faced some issues. I am attaching the screenshot below:

- Added thread group -> Renamed the thread group as User
- There are set to be 10 users and 2 loop counts.



Thread Group

Name: User

Comments:

Action to be taken after a Sampler error

● Continue ○ Start Next Thread Loop ○ Stop Thread ○ Stop Test ○ Stop Test Now

Thread Properties

Number of Threads (users): 10

Ramp-up period (seconds): 1

Loop Count: □ Infinite 2

☑ Same user on each iteration

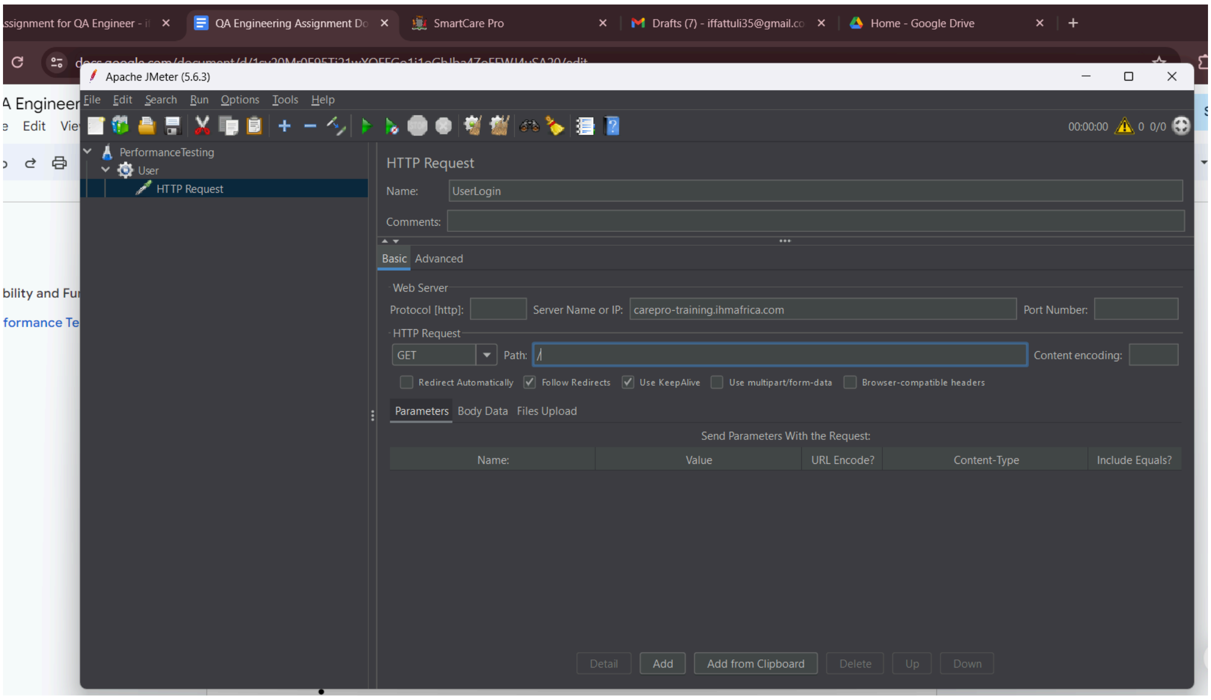
□ Delay Thread creation until needed

□ Specify Thread lifetime

Duration (seconds):

Startup delay (seconds):

- Add>Sampler>Http Request—> Renamed AS UserLogin
- Added the url of login page in the Server Name or IP field by removing the last '/' and first 'https://'.
- Added the last '/' in the Path field.



HTTP Request

Name: UserLogin

Comments:

Basic Advanced

Web Server

Protocol [http]: Server Name or IP: carepro-training.ihmafrica.com Port Number:

HTTP Request

GET Path: / Content encoding:

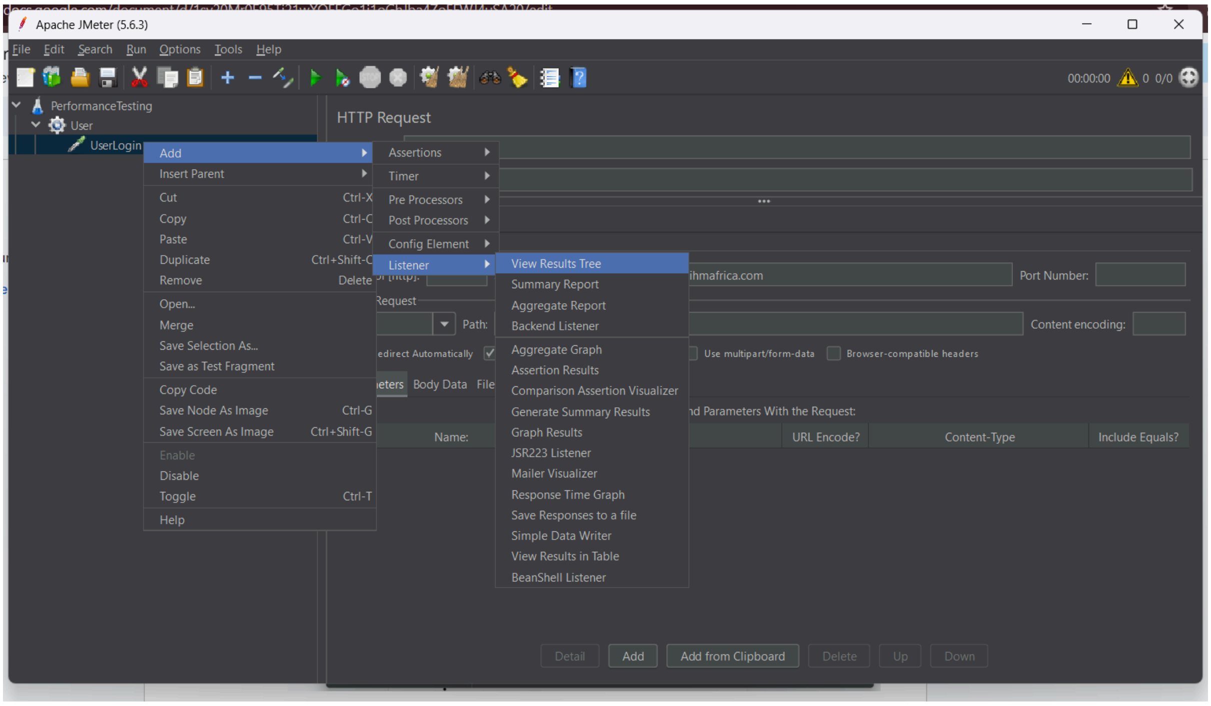
□ Redirect Automatically ☑ Follow Redirects ☑ Use KeepAlive □ Use multipart/form-data □ Browser-compatible headers

Parameters Body Data Files Upload

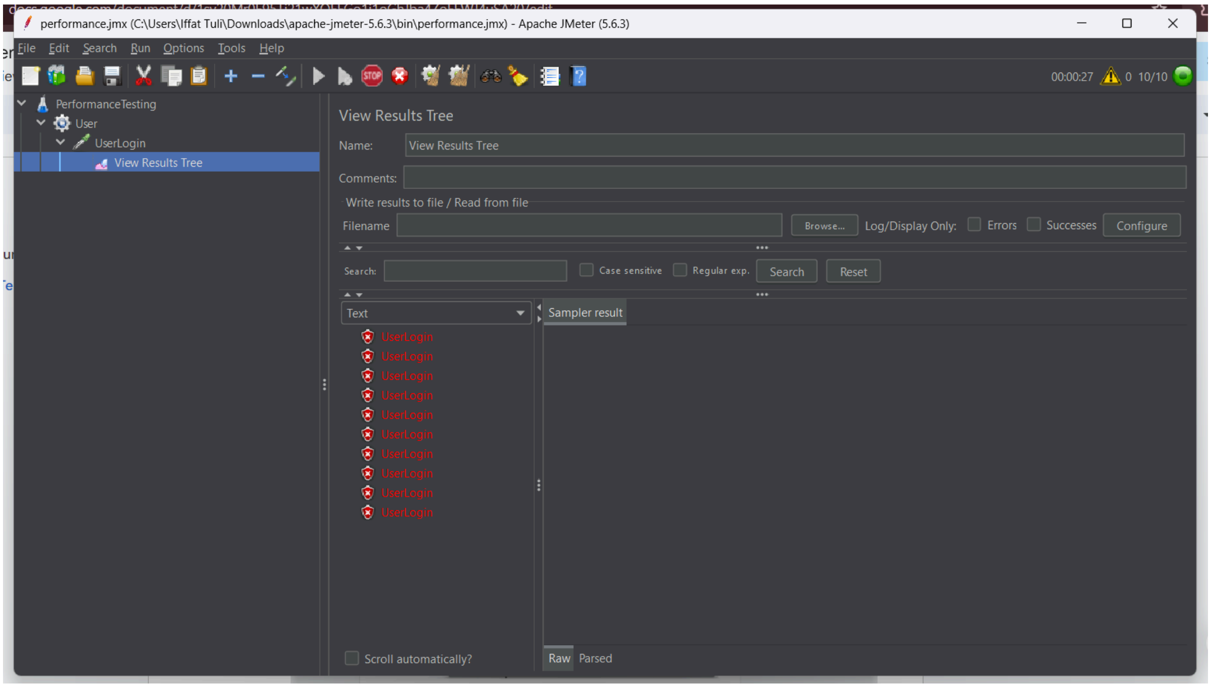
Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?

- Added view results tree listener for the report



- The report is attached below:


performance.jmx (C:\Users\Iffat Tuli\Downloads\apache-jmeter-5.6.3\bin\performance.jmx) - Apache JMeter (5.6.3)
File Edit Search Run Options Tools Help
00:00:27 0 10/10
PerformanceTesting
User
UserLogin
View Results Tree
View Results Tree
Name: View Results Tree
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only:
Errors
Successes
Configure
Search:
Case sensitive
Regular exp.
Search
Reset
Text
Sampler result
UserLogin
UserLogin
UserLogin
UserLogin
UserLogin
UserLogin
UserLogin
UserLogin
UserLogin
UserLogin
Scroll automatically?
Raw
Parsed