Apollo 24/7 Web Application - Test Report  
By: Saravana Sakthi P

Date: 13-09-2025

# Abstraction

The project is an Apollo Healthcare Web Application designed to facilitate patients in accessing healthcare services online. It provides features such as launching the browser to access the system, secure login with mobile and OTP authentication, searching for doctors by specialty, date, and location, viewing detailed doctor profiles, scheduling appointments, adding patient details, and confirming appointments. The application ensures both usability and security by handling invalid inputs with appropriate error messages and validating correct workflows for successful operations.  
  
Purpose of the Overall Project  
The purpose of the Apollo Healthcare Web Application is to provide patients with an easy, secure, and efficient way to access healthcare services online.   
It supports login, doctor search, appointment scheduling, patient management, and other features to enhance user experience.  
  
Outcome of the Overall Project  
The outcome of testing confirms that the application is stable, user-friendly, and secure. All major functions like login, doctor search, appointment scheduling, and patient management work as expected, with proper validations and error handling.

# Feature: Launch Browser

**Purpose while testing application:** To validate that users can successfully launch the browser and access the Apollo home page for further actions.

**Outcome while testing application:** The browser launches, navigates to the website, and loads the homepage without issues.  
Here is the GitHub link for this feature : <https://github.com/Sak-12s/Sprint-Implementation-Apollo-24-7-Project/blob/main/src/test/resources/Features/1_launchBrowser.feature>

## Scenario: Launch browser successfully

**Purpose**: To confirm that the browser launches properly and the homepage is displayed without errors.

**Outcome:** The user is able to open the website and access the Apollo homepage.

### Test Case Details:

**Steps**: Launch browser → Open website

**Expected Result:** The Apollo homepage loads successfully

**Actual Result:** The Apollo homepage got loaded successfully

**Status**: Pass

# Feature: Login

**Purpose while testing application:** To test the system’s authentication mechanism, ensuring that valid users can log in while invalid entries are rejected with appropriate error messages.

**Outcome while testing application:** The system successfully authenticates users with correct credentials and prevents access with invalid details.  
Here is the GitHub link for this feature : <https://github.com/Sak-12s/Sprint-Implementation-Apollo-24-7-Project/blob/main/src/test/resources/Features/2_login.feature>

## Scenario: Verify login failure with invalid mobile number

**Purpose**: To ensure that the system rejects login attempts with an invalid mobile number.

**Outcome**: The system displays an error message stating “Invalid mobile number.”

### Test Case Details:

**Steps**: Enter invalid mobile number → Attempt login

**Expected Result**: Error message appears

**Actual Result**: Error message is appeared

**Status**: Pass

## Scenario: Verify login failure with invalid OTP

**Purpose**: To confirm that incorrect OTP entries are rejected during login.

**Outcome**: The system displays “Invalid OTP” when an incorrect OTP is submitted.

### Test Case Details:

**Steps**: Enter valid mobile number → Enter invalid OTP → Submit

**Expected Result**: Error message “Invalid OTP” is shown

**Actual Result**: Error message “Invalid OTP” is displayed

**Status**: Pass

## Scenario: Login successfully

**Purpose**: To verify that the system allows users to log in with valid credentials.

**Outcome**: The user is authenticated and redirected to the application’s homepage.

### Test Case Details:

**Steps**: Enter valid mobile number → Request OTP → Enter correct OTP → Submit

**Expected Result**: User logs in successfully

**Actual Result**: User logged in successfully

**Status**: Pass

# Feature: Profile Addition

**Purpose while testing application**: To ensure that users can input and save profile details securely and that validations are in place for required fields.

**Outcome while testing application**: User profile information is saved after proper validation and incorrect inputs trigger appropriate error messages.  
Here is the GitHub link for this feature : <https://github.com/Sak-12s/Sprint-Implementation-Apollo-24-7-Project/blob/main/src/test/resources/Features/3_profileAddition.feature>

## Scenario: Verify profile addition successfully

**Purpose**: To confirm that profile details are added without errors and stored in the system.

**Outcome**: The profile is created successfully, and a confirmation message is displayed.

### Test Case Details:

**Steps**: Enter first name → Enter last name → Enter DOB → Select profile preferences → Select gender → Enter email → Click submit

**Expected Result**: Profile is saved and confirmation message is displayed

**Actual Result**: Profile got saved and confirmation message is displayed

**Status**: Pass

# Feature: Set Notification Preferences

**Purpose while testing application**: To allow users to configure their notification settings and ensure that the system saves these preferences.

**Outcome while testing application**: Notification preferences are applied as per user selection and users receive notifications accordingly.  
Here is the GitHub link for this feature : <https://github.com/Sak-12s/Sprint-Implementation-Apollo-24-7-Project/blob/main/src/test/resources/Features/4_setNotificationPreferences.feature>

## Scenario: Verify notification preferences setting successfully

**Purpose**: To ensure that users can select and save notification preferences without issues.

**Outcome**: The system applies the preferences and confirms the changes.

### Test Case Details:

**Steps**: Navigate to notification preferences → Select desired options → Save settings

**Expected Result**: Preferences are applied successfully

**Actual Result**: Preferences got applied successfully

**Status**: Pass

# Feature: Viewing Premium Insurance Plans

**Purpose while testing application**: To ensure that users can explore insurance plans and make informed decisions based on the available information.

**Outcome while testing application:** Users can browse and view insurance plan details without technical errors.  
Here is the GitHub link for this feature : <https://github.com/Sak-12s/Sprint-Implementation-Apollo-24-7-Project/blob/main/src/test/resources/Features/5_viewingPremiumInsurancePlans.feature>

## Scenario: Verify viewing insurance plans successfully

**Purpose**: To confirm that insurance plans are accessible and their details are displayed correctly.

**Outcome**: The system displays available insurance plans with accurate information.

### Test Case Details:

**Steps**: Navigate to insurance plans → View details

**Expected Result**: Insurance plans and their details are shown

**Actual Result**: Insurance plans and their details are displayed

**Status**: Pass

# Feature: Weight Management

**Purpose while testing application**: To ensure that users can input, update, and view their weight history for effective health tracking.

**Outcome while testing application**: Users can log and monitor their weight, and the system displays historical data for review.  
Here is the GitHub link for this feature : <https://github.com/Sak-12s/Sprint-Implementation-Apollo-24-7-Project/blob/main/src/test/resources/Features/6_weightMangement.feature>

## Scenario: Verify weight tracking functionality

**Purpose**: To confirm that the weight management feature accepts inputs and displays accurate tracking history.

**Outcome**: Weight entries are saved and shown as a log for user reference.

### Test Case Details:

**Steps**: Navigate to weight management → Enter weight → Save → View history

**Expected Result**: Weight data is recorded and displayed

**Actual Result**: Weight data is recorded and BMI is displayed

**Status**: Pass

# Feature: Logout

**Purpose while testing application**: To ensure that users can log out safely and are redirected to the login page without data leakage.

**Outcome while testing application**: The logout process ends the session and redirects users as expected.  
Here is the GitHub link for this feature : <https://github.com/Sak-12s/Sprint-Implementation-Apollo-24-7-Project/blob/main/src/test/resources/Features/7_logout.feature>

## Scenario: Verify logout functionality

**Purpose**: To validate that the logout process is executed securely and without issues.

**Outcome**: The session is terminated and the login page is displayed.

### Test Case Details:

**Steps**: Click logout → Confirm logout → Redirect to login page

**Expected Result**: Session ends and user is redirected

**Actual Result**: Session ends and user is logged out of the website

**Status**: Pass  
  
  
Conclusion of the whole project  
  
The Apollo Healthcare Web Application successfully meets its objectives by providing a secure, efficient, and user-friendly platform for accessing healthcare services. All key functionalities—including login, profile addition, notification settings, viewing insurance plans, and weight management—have been thoroughly tested and validated to work smoothly. The application effectively handles both valid and invalid inputs, ensuring data security and a seamless user experience. The results of the testing confirm that the application is ready for deployment, with opportunities for further enhancements to improve patient care and engagement.