Test Case Development for eChannelling Website

1. **1. Test Cases for Important Test Scenarios**

Based on the test scenarios identified in Assignment 1, here are five detailed test cases for the most critical functionality of the eChannelling website:

Test Case 1: User Registration Process

Description: Verify a new user can successfully register on the eChannelling platform
Priority: High
Preconditions:

- Website is accessible
- Test user account does not already exist

Test Steps:

2. Navigate to the eChannelling website homepage
3. Click on the "Sign Up" or "Register" option
4. Enter valid details in all required fields:
   - Full name: "Nethmi Kaveesha"
   - Email: "nethmikaveesha@gmail.com"
   - Mobile number: "0712345678"
   - Password: "Test@1234"
   - Confirm password: "Test@1234"
5. Accept terms and conditions
6. Click "Register" button

Expected Results:

- Success message is displayed
- Confirmation email/SMS is sent to the provided contact details
- User is redirected to the dashboard or login page
- A new account is created in the system

Post-conditions:

- Users can log in with new credentials
- User profile is stored in the database

Test Case 2: Appointment Booking Process

Description: Verify user can search for a doctor and successfully book an appointment
Priority: Critical
Preconditions:

- User is logged in
- At least one doctor is available for booking

Test Steps:

1. Navigate to "Book Appointment" or "Find Doctor" section
2. Search for a doctor by specialty (e.g., "Cardiology")
3. Select a doctor from search results
4. Choose an available date and time slot
5. Confirm patient details
6. Select payment method (credit card)
7. Enter payment details
8. Confirm booking

Expected Results:

- Appointment is successfully booked
- Confirmation page displays appointment details and reference number
- Confirmation SMS/email is sent to user
- The selected time slot becomes unavailable for other users
- Payment is properly processed

Post-conditions:

- Appointment appears in user's appointment history
- Doctor's schedule is updated

Test Case 3: Login Authentication with Valid Credentials

Description: Verify user can login with correct username and password
Priority: High
Preconditions:

- User account exists in the system
- User is on the login page

Test Steps:

1. Enter valid email "nethmikaveesha@gmail.com"

2. Enter correct password "Test@1234"
3. Click "Login" button

Expected Results:

- User is successfully authenticated
- User is redirected to dashboard/home page
- User's name is displayed in the header/profile section
- All authenticated features become accessible

Post-conditions:

- Session is created for the user
- Login activity is logged in system

Test Case 4: Viewing Appointment History

Description: Verify user can access and view their appointment history
Priority: Medium
Preconditions:

- User is logged in
- User has at least one booked appointment

Test Steps:

1. Navigate to user dashboard
2. Click on "My Appointments" or "Channel History" option
3. Review the list of appointments

Expected Results:

- List of user's appointments is displayed
- Appointments show correct details (doctor name, hospital, date/time, status)
- List is filtered to show only the logged-in user's appointments
- Upcoming appointments may be highlighted or sorted first

Post-conditions:

- No system state change (read-only operation)

Test Case 5: Security Validation for Invalid Login

Description: Verify system handles invalid login attempts securely
Priority: High
Preconditions:

- Valid user account exists
- User is on login page

Test Steps:

1. Enter valid username "nethmikaveesha@gmail.com"
2. Enter incorrect password "WrongPassword123"
3. Click "Login" button
4. Observe system response

Expected Results:

- Login fails
- Error message displayed ("Invalid username or password")
- User remains on login page
- No sensitive information is revealed in error message
- User can try again

Post-conditions:

- Failed login attempt may be logged (for security monitoring)
- If implemented, account lockout counter may be incremented

7. **2. Reasons Why Software Quality Engineers Cannot Perform Unit Testing**

As a Software Quality Engineer, there are several fundamental limitations that prevent conducting effective unit testing:

1. **Lack of Access to Source Code**: Software Quality Engineers typically don't have the same level of access to the application's source code as developers. Unit testing requires intimate knowledge of internal methods, classes, and code structure, which QA engineers often cannot access directly. For eChannelling, the backend code that processes appointment bookings would be inaccessible to QA.
2. **No Development Environment Setup**: Unit testing requires a specialized development environment with appropriate compilers, building tools, and integrated testing frameworks that are set up for the specific technology stack. QA engineers usually work in testing environments rather than development environments, making it difficult to execute unit tests for eChannelling's core components.
3. **Limited Programming Knowledge of Implementation Language**: Effective unit testing requires in-depth knowledge of the programming language and framework used for development. If eChannelling is built with technologies like Java Spring Boot or Node.js, QA engineers might not possess the specialized programming expertise to write meaningful unit tests for these frameworks.

4. **Ownership and Responsibility Boundaries**: Unit testing is fundamentally a developer responsibility within the software development lifecycle. Developers who write the code have the knowledge to test individual units of functionality. QA engineers focus on integration, system, and acceptance testing rather than testing individual code components. This separation of responsibilities is a standard practice in software development.
5. **Timing in Development Cycle**: Unit testing occurs during the development phase, typically before code is committed to shared repositories. By the time Software Quality Engineers receive the application for testing, the code has already progressed beyond the unit testing stage. The QA role begins when components are already integrated, making unit testing redundant and potentially disruptive to the development workflow.

8. **3. Automated Testing Tools for eChannelling Website**

a. Criteria to Evaluate Automated Testing Tools

When selecting automated testing tools for the eChannelling website, the following criteria are essential for evaluation:

1. **Cross-Browser and Cross-Device Compatibility**: The tool must support testing across multiple browsers (Chrome, Firefox, Safari, Edge) and device types (desktop, tablet, mobile) to ensure eChannelling works consistently for all users regardless of their access method.
2. **Integration Capabilities**: The tool should integrate seamlessly with existing development and CI/CD pipelines, bug tracking systems (like Jira), and test management tools. Integration with technologies likely used by eChannelling (such as Jenkins, Git, or Azure DevOps) is particularly important.
3. **Support for Key Technologies**: Given that eChannelling likely uses a combination of frontend (HTML, CSS, JavaScript) and backend technologies, the tool must support testing of these technologies, especially for critical functionalities like appointment booking and payment processing.
4. **Ease of Test Creation and Maintenance**: The tool should offer features like record-and-playback, visual test creation, or low-code interfaces that reduce the learning curve while providing options for more advanced scripting when needed. Test maintenance should be efficient to handle eChannelling's frequent updates.
5. **Cost-Effectiveness and ROI**: The total cost of ownership including licensing, infrastructure, training, and maintenance should be evaluated against the expected benefits in terms of increased test coverage, reduced manual testing effort, and improved quality of the eChannelling platform.

b. Research on Automated Testing Tools

Based on the identified criteria, here's an analysis of ten automated testing tools:

1. Selenium

- **Cross-Browser/Device Compatibility**: Excellent support for all major browsers and can be configured for mobile testing through Appium
- **Integration Capabilities**: Integrates with most CI/CD tools, test management systems, and supports multiple programming languages

- **Technology Support**: Strong support for web technologies including JavaScript, HTML5, and dynamic content
- **Ease of Use**: Moderate to complex; requires programming knowledge but offers flexibility
- **Cost**: Open-source core with free usage, though implementation and maintenance costs can be significant
- **Suitability for eChannelling**: High for web testing but requires significant development expertise

2. Katalon Studio

- **Cross-Browser/Device Compatibility**: Supports multiple browsers and devices through built-in integrations
- **Integration Capabilities**: Good integration with CI/CD tools, Jira, and test management platforms
- **Technology Support**: Built on Selenium with extended support for web, API, and mobile testing
- **Ease of Use**: More user-friendly than Selenium with record-and-playback plus customization options
- **Cost**: Free version available with paid enterprise options
- **Suitability for eChannelling**: Good balance of capabilities and ease of use for healthcare portal testing

3. TestComplete

- **Cross-Browser/Device Compatibility**: Comprehensive browser coverage with desktop and mobile testing capabilities
- **Integration Capabilities**: Integrates with DevOps tools and various frameworks
- **Technology Support**: Supports various technologies including web, desktop, and mobile applications
- **Ease of Use**: User-friendly interface with record-and-playback plus scripting capabilities
- **Cost**: Commercial tool with licensing fees
- **Suitability for eChannelling**: Strong for complex testing scenarios but may be costly

4. Cypress

- **Cross-Browser/Device Compatibility**: Initially Chrome-only but now supports most browsers; limited mobile testing
- **Integration Capabilities**: Modern integrations with CI/CD pipelines and development tools
- **Technology Support**: Excellent for modern web applications, especially JavaScript frameworks
- **Ease of Use**: Developer-friendly with good debugging capabilities but steeper learning curve for non-developers

- **Cost**: Open-source core with paid dashboard for larger teams
- **Suitability for eChannelling**: Excellent for modern web front-end testing but limited for mobile

5. Playwright

- **Cross-Browser/Device Compatibility**: Excellent cross-browser support including Chromium, Firefox, and Safari
- **Integration Capabilities**: Good integration with modern development stacks and CI/CD tools
- **Technology Support**: Strong support for modern web technologies and frameworks
- **Ease of Use**: Requires programming knowledge but offers powerful and concise APIs
- **Cost**: Open-source and free to use
- **Suitability for eChannelling**: Good for testing modern web applications with complex interactions

6. UFT (Unified Functional Testing)

- **Cross-Browser/Device Compatibility**: Strong support for browsers and devices
- **Integration Capabilities**: Extensive integration with HP ALM and other enterprise tools
- **Technology Support**: Comprehensive technology support including web, desktop, and API
- **Ease of Use**: Relatively user-friendly but requires training
- **Cost**: High commercial licensing fees
- **Suitability for eChannelling**: Strong enterprise option but expensive for smaller operations

7. TestProject

- **Cross-Browser/Device Compatibility**: Good browser and device coverage
- **Integration Capabilities**: Integrates with popular development and testing tools
- **Technology Support**: Built on Selenium and Appium with additional capabilities
- **Ease of Use**: Very user-friendly with cloud-based management
- **Cost**: Free with limitations, paid enterprise options
- **Suitability for eChannelling**: Good for teams without extensive programming experience

8. Appium

- **Cross-Browser/Device Compatibility**: Primarily for mobile testing, but works with mobile web apps
- **Integration Capabilities**: Integrates with standard testing frameworks and CI/CD systems
- **Technology Support**: Strong for mobile applications and mobile web testing
- **Ease of Use**: Moderate to complex, requires programming knowledge

- **Cost**: Open-source and free
- **Suitability for eChannelling**: Good specifically for mobile testing aspects of the website

9. Tricentis Tosca

- **Cross-Browser/Device Compatibility**: Comprehensive coverage across browsers and devices
- **Integration Capabilities**: Strong enterprise integrations including test management and requirements
- **Technology Support**: Broad technology support including legacy systems
- **Ease of Use**: Model-based approach that's easier for non-programmers
- **Cost**: High commercial licensing fees
- **Suitability for eChannelling**: Good for large-scale enterprise implementation

10. Ranorex

- **Cross-Browser/Device Compatibility**: Good browser and device coverage
- **Integration Capabilities**: Integrates with common development and CI/CD tools
- **Technology Support**: Supports web, desktop, and mobile testing
- **Ease of Use**: User-friendly interface with both codeless and code-based options
- **Cost**: Commercial tool with licensing fees
- **Suitability for eChannelling**: Good balance of usability and features for mid-size implementation

c. Recommended Automated Testing Tools

After evaluating the ten automated testing tools against our defined criteria, the following two are recommended as the best suited for the eChannelling website:

1. Katalon Studio

Justification:

- **Comprehensive Coverage**: Katalon provides excellent cross-browser and cross-device testing capabilities, essential for eChannelling's diverse user base accessing from different devices and browsers.
- **Balance of Accessibility and Power**: It offers both codeless approaches for QA team members with less programming experience and scripting options for complex scenarios. This flexibility makes it suitable for testing eChannelling's various components from simple registrations to complex appointment bookings and payment processing.
- **Healthcare-Specific Suitability**: Katalon's handling of form submissions, data-driven testing, and API testing capabilities align perfectly with eChannelling's needs as a healthcare appointment platform with sensitive data requirements.

- **Cost-Effective Implementation**: With a free tier that offers significant functionality and reasonably priced enterprise options, Katalon provides good ROI without prohibitive initial investment. This makes it feasible to implement within the typical healthcare IT budgets.
- **Reduced Learning Curve**: The built-in record-and-playback feature combined with the ability to enhance scripts manually provides an ideal balance for the QA team to quickly create and maintain tests for eChannelling's frequent updates.

2. Playwright

Justification:

- **Modern Web Application Focus**: Playwright excels at testing modern web applications with dynamic content and complex user interactions, which aligns perfectly with eChannelling's appointment booking system and payment processes.
- **Superior Performance**: Its architecture provides faster and more reliable test execution compared to traditional tools, essential for comprehensive regression testing of eChannelling's critical patient-facing features.
- **Strong Authentication Testing**: Playwright's capabilities for handling multiple browser contexts and states make it particularly well-suited for testing the security aspects of eChannelling, including login processes and session management.
- **Comprehensive API Testing**: The ability to combine UI testing with API testing in the same framework is valuable for ensuring that eChannelling's frontend and backend systems work together correctly, especially for critical features like appointment scheduling.
- **Future-Proofing**: As eChannelling evolves to incorporate more modern web technologies, Playwright's excellent support for the latest web standards and frameworks ensures the testing solution will remain relevant and effective.
- **Cost-Efficient**: Being open source with no licensing fees, Playwright offers excellent capabilities without ongoing license costs, though it does require more technical expertise to implement.

These two tools provide complementary strengths that together address all the testing needs of the eChannelling platform. Katalon Studio offers an accessible yet powerful approach suitable for a range of team skill levels, while Playwright provides cutting-edge capabilities for testing modern web applications with complex interactions. By implementing both tools, the eChannelling QA team can ensure comprehensive test coverage across different testing scenarios and technical requirements.