**1. Authentication**

* **Feature:** User Authentication
* **Goal:** Ensure users can securely log in and their roles are correctly assigned.
* **Scenario 1: Successful Login**
  + Given the user is on the login page
  + When the user enters a valid username and password
  + And clicks the "Login" button
  + Then the user should be redirected to the1 main application page
  + And the user should be authenticated as a specific role (e.g., "Editor")
* **Scenario 2: Invalid Login - Incorrect Password**
  + Given the user is on the login page
  + When the user enters a valid username and an incorrect password
  + And clicks the "Login" button
  + Then the user should see an error message indicating invalid credentials
  + And the user should remain on the login page
* **Scenario 3: Invalid Login - Incorrect Username**
  + Given the user is on the login page
  + When the user enters an invalid username and a valid password
  + And clicks the "Login" button
  + Then the user should see an error message indicating invalid credentials
  + And the user should remain on the login page
* **Scenario 4: Admin User Login**
  + Given the user is on the login page
  + When the user enters the username and password for an "Admin" user
  + And clicks the "Login" button
  + Then the user should be redirected to the main application page
  + And the user should have access to admin-specific features (e.g., user management)
* **Scenario 5: Viewer User Login**
  + Given the user is on the login page
  + When the user enters the username and password for a "Viewer" user
  + And clicks the "Login" button
  + Then the user should be redirected to the main application page
  + And the user should have read-only access to client data

**2. Client Stack Management**

* **Feature:** Creating a New Client Stack
* **Goal:** Verify that users can create new client stack records with the correct data structure.
* **Scenario 1: Successful Client Stack Creation**
  + Given the user is logged in as an "Editor"
  + And the user is on the client list page
  + When the user clicks the "Create New Client" button
  + And the user enters all required client details (name, etc.)
  + And the user enters stack details using the template (tiers, etc.)
  + And the user clicks the "Save" button
  + Then a new client record should be created in the database
  + And the user should be redirected to the new client's detail page
  + And the new client's details should be displayed correctly
* **Scenario 2: Client Stack Creation with Missing Required Fields**
  + Given the user is logged in as an "Editor"
  + And the user is on the client list page
  + When the user clicks the "Create New Client" button
  + And the user enters some client details but leaves out a required field (e.g., client name)
  + And the user clicks the "Save" button
  + Then an error message should be displayed indicating the missing field
  + And the client record should not be created
* **Scenario 3: Client Stack Creation with Invalid Data**
  + Given the user is logged in as an "Editor"
  + And the user is on the client list page
  + When the user clicks the "Create New Client" button
  + And the user enters data in an incorrect format (e.g., invalid date)
  + And the user clicks the "Save" button
  + Then an error message should be displayed indicating the invalid data
  + And the client record should not be created
* **Feature:** Editing an Existing Client Stack
* **Goal:** Ensure users can modify client stack data and that changes are saved correctly.
* **Scenario 1: Successful Client Stack Edit**
  + Given the user is logged in as an "Editor"
  + And the user is on a client's detail page
  + When the user modifies client details (e.g., adds a tag)
  + And the user clicks the "Save" button
  + Then the client record should be updated in the database
  + And the updated details should be displayed on the client's detail page
* **Scenario 2: Editing Client Stack with Invalid Data**
  + Given the user is logged in as an "Editor"
  + And the user is on a client's detail page
  + When the user modifies client details with invalid data
  + And the user clicks the "Save" button
  + Then an error message should be displayed
  + And the client record should not be updated with the invalid data
* **Scenario 3: Editing Client Stack with No Changes**
  + Given the user is logged in as an "Editor"
  + And the user is on a client's detail page
  + When the user does not modify any client details
  + And the user clicks the "Save" button
  + Then the client record should not be updated
  + And a message should be displayed indicating no changes were made (optional)
* **Feature:** Deleting a Client Stack
* **Goal:** Verify that users with the correct permissions can delete client stack records.
* **Scenario 1: Successful Client Stack Deletion**
  + Given the user is logged in as an "Admin"
  + And the user is on a client's detail page
  + When the user clicks the "Delete Client" button
  + And confirms the deletion
  + Then the client record should be deleted from the database
  + And the user should be redirected to the client list page
  + And the deleted client should not be visible in the list
* **Scenario 2: Client Stack Deletion Attempt by Non-Admin**
  + Given the user is logged in as an "Editor"
  + And the user is on a client's detail page
  + When the user attempts to delete the client stack
  + Then the delete option should not be available or should be disabled
  + Or the user should receive an authorization error message

**3. Data Visualization**

* **Feature:** Displaying Client Stack Data
* **Goal:** Ensure client stack data is displayed in both tiered and mind map formats.
* **Scenario 1: Displaying Tiered View**
  + Given the user is logged in
  + And the user is on a client's detail page
  + When the user selects the "Tiered View"
  + Then the client's stack data should be displayed in a structured, tier-based format
* **Scenario 2: Displaying Mind Map View**
  + Given the user is logged in
  + And the user is on a client's detail page
  + When the user selects the "Mind Map View"
  + Then the client's stack data should be displayed in an unstructured, mind map-style graphical display
* **Scenario 3: Displaying Client Stack with Nested Tiers**
  + Given the user is logged in
  + And the user is on a client's detail page
  + When the client's stack data includes nested tiers
  + Then the tiered view should display the nested tiers correctly
  + And the mind map view should represent the nested tiers in its graphical display

**4. Search and Filtering**

* **Feature:** Searching for Clients
* **Goal:** Verify that users can search for clients based on various criteria.
* **Scenario 1: Successful Client Search**
  + Given the user is logged in
  + And the user is on the client list page
  + When the user enters a client's name in the search bar
  + And clicks the "Search" button
  + Then the client list should be filtered to display only matching clients
* **Scenario 2: Client Search with No Results**
  + Given the user is logged in
  + And the user is on the client list page
  + When the user enters a non-existent client name in the search bar
  + And clicks the "Search" button
  + Then the client list should display a message indicating no matching clients were found
* **Scenario 3: Client Search by Tag**
  + Given the user is logged in
  + And the user is on the client list page
  + When the user searches using a client tag
  + Then the client list should be filtered to show clients with that tag
* **Feature:** Filtering Clients
* **Goal:** Verify that users can filter clients based on criteria
* **Scenario 1: Filter Clients by Flags**
  + Given the user is logged in
  + And the user is on the client list page
  + When the user filters clients by a specific flag (e.g., "Renewal Due")
  + Then the client list should only display clients with that flag

**5. Data Export**

* **Feature:** Exporting Client Stack Data
* **Goal:** Ensure client stack data can be exported in different formats.
* **Scenario 1: Export to PDF - Mind Map**
  + Given the user is logged in
  + And the user is on a client's detail page
  + When the user selects "Export to PDF" and chooses the "Mind Map" option
  + Then a PDF file should be generated
  + And the PDF file should contain the client's stack data in a mind map format
* **Scenario 2: Export to PDF - Tiered List**
  + Given the user is logged in
  + And the user is on a client's detail page
  + When the user selects "Export to PDF" and chooses the "Tiered List" option
  + Then a PDF file should be generated
  + And the PDF file should contain the client's stack data in a tiered list format
* **Scenario 3: Export to CSV**
  + Given the user is logged in
  + And the user is on a client's detail page
  + When the user selects "Export to CSV"
  + Then a CSV file should be generated
  + And the CSV file should contain the client's stack data in a flat data table format
* **Scenario 4: Export with Hidden Fields**
  + Given the user has hidden certain fields in the UI
  + When the user exports data
  + Then the exported file should not include the hidden fields
* **Scenario 5: Export with Tags and Flags Toggled Off**
  + Given the user is on the export settings
  + When the user toggles off the "tags" and "flags" options
  + And the user exports the data
  + Then the exported file should not include tags and flags

**6. Data Input**

* **Feature:** Manual Data Entry
* **Goal:** Verify that users can manually input client stack data.
* **Scenario 1: Successful Data Entry**
  + Given the user is logged in as an "Editor"
  + And the user is on the client stack input form
  + When the user enters valid data into all the fields
  + And clicks "Save"
  + Then the data is saved correctly.
* **Scenario 2: Data Entry with Freeform Text**
  + Given the user is logged in as an "Editor"
  + And the user is on the client stack input form
  + When the user enters freeform text into fields like "NextGen Firewall" (e.g., "Palo Alto")
  + And clicks "Save"
  + Then the data is saved correctly.
* **Scenario 3: Inline Editing**
  + Given the user is logged in as an "Editor"
  + And the user is viewing a client stack
  + When the user edits data directly within the display
  + And saves the changes
  + Then the data is updated correctly.

**7. User Roles and Permissions**

* **Feature:** Access Control
* **Goal:** Verify that user roles and permissions are enforced correctly.
* **Scenario 1: Admin User Access**
  + Given the user is logged in as an "Admin"
  + Then the user should have access to all features, including user management, client data editing, and export functionality
* **Scenario 2: Editor User Access**
  + Given the user is logged in as an "Editor"
  + Then the user should be able to create and update client data records
  + And the user should not have access to admin-specific features
* **Scenario 3: Viewer User Access**
  + Given the user is logged in as a "Viewer"
  + Then the user should have read-only access to client data
  + And the user should not be able to create, update, or delete client records

**8. Data Management**

* **Feature:** Data Backup
* **Goal:** Verify that the database can be backed up.
* **Scenario 1: Database Backup**
  + Given the user is logged in as an "Admin"
  + When the user initiates a database backup
  + Then the database should be exported as a zipped JSON or CSV file

**9. Change Log**

* **Feature:** Change Log Tracking
* **Goal:** Verify that changes to client stacks are logged.
* **Scenario 1: Change Log Entry on Edit**
  + Given the user is logged in as an "Editor"
  + And the user edits a client stack
  + When the user saves the changes
  + Then a new entry should be added to the change log
  + And the change log entry should include user identity, timestamp, and action description

**10. UI/UX**

* **Feature:** Responsive Design
* **Goal:** Verify the UI is responsive.
* \*\*Scenario 1: Mobile Responsiveness
  + Given the user is on a mobile device (e.g., iPhone 15 Pro Max)
  + When the user accesses the application
  + Then the UI should adapt to the device's screen size and orientation
  + And the UI should be touch-first.
* **Feature:** Dark Mode
* **Goal:** Verify dark mode functionality.
* \*\*Scenario 1: Persistent Dark Mode
  + Given the user toggles dark mode on
  + When the user navigates through the application
  + Then dark mode should persist across all pages and sessions