

Test Identifier	Test Summary	Test Description	Precondition	Test Steps	Expected Result	Actual Results	UI / API	Status	Tester	Defect ID
UI/TC01	Verify Search plants list by Plant Name as an admin and display the searched list.	Check searching for an existing plant by name	1. User login as an admin 2. Want have Plants in Plant list 3. Admin user is on "ui/plants"	1. Enter existing Plant name in the Search input field 2. click the search button	• Want to display the matching plants according to the search name's result		UI	Not Executed	Dil rangi 214102J	
UI/TC02	Verify Search plants list by Plant Name with space as an admin and display the searched list.	Check searching for an existing plant by name with space	1. User login as an admin 2. Want have Plants in Plant list 3. Admin user is on "ui/plants"	1. Enter existing Plant name and add space in the Search input field 2. click the search button	• Want to display the matching plants according to the search name's result		UI	Not Executed	Dil rangi 214102J	
UI/TC03	Verify "Low" Stock Badge Visibility by user as an admin	When the plant stock quantity is less than 5, check the visibility of Low badge indicator	1. User login as an admin 2. want to have plants in the list with stock quantity less than 5 3. admin user is on "ui/plants"	1)click add plant button 2)input valid values to create new plant add quantity less than 5	• want to display the low badge for the plants that have quantity less than 5		UI	Not Executed	Dil rangi 214102J	
UI/TC04	Verify Action column Visibility on Admin	Validates that the Admin has full management controls over the inventory. It ensures that "Edit" and "Delete" options are available for every row, allowing the Admin to modify or remove plant records.	1. The user needs to log in as an admin user. 2. The admin user is on the Plant List page. 3. At least one plant exists in the list.	1. Observe the rows in the plant list table. 2. Check the "Actions" column for any specific plant row.	• The "Edit" (pencil icon) button is visible.  • The "Delete" (trash/bin icon) button is visible for the selected row.		UI	Not Executed	Dil rangi 214102J	
UI/TC05	Verify Edit Plant Functionality	Check that Admin can change the Existing details of an existing plant	1. User login as an admin 2. want to have at least one plant in plants list 3. admin user is on "ui/plants/add"	1. Click "Edit" button on a plant row. 2. Update the "Input fields". 3. Click "Save".	• System redirects to Plant list, all updated details want to be updated in the table.		UI	Not Executed	Dil rangi 214102J	
UI/TC06	Verify Delete Plant Functionality	Check that Admin can Delete an existing plant	1. User login as an admin 2. want to have at least one plant in plants list 3. admin user is on "ui/plants"	1. Click "Delete" button on a plant row. 2. Click "ok" when popup".	• System redirects to Plant list, and deleted plant not be in the plant list.		UI	Not Executed	Dil rangi 214102J	

UI/TC07	Verify Cancel Button Functionality in Plant Add page	Check that Admin can cancel the an creating plant	1. User login as an admin 2. admin user is on "/ui/plants/add"	1. Enter data to the input fields 2. Click "Cancel" button on a plant row.	• The plant creation is canceled and navigate to the Plant list "/ui/plants".		UI	Not Executed	Dilrangi 214102J	
UI/TC08	Verify the category list is Sub category list	Check that category dropdown display the only sub categories	1. User login as an admin 2. admin user is on "/ui/plants/add"	1. click the dropdown of sub category	• Want to list only the sub categories		UI	Not Executed	Dilrangi 214102J	
UI/TC09	Verify Plant List Visibility User as non admin	Validates that a standard User can successfully navigate to and view the Plant Management module.	1. The user needs to log in as a Standard User (Non-Admin).  2. The user is on the Dashboard.	1. Click on the "Plants" link in the navigation menu.  2. Observe the central content area.	• The system redirects to the Plant List page.  • The plant list table is displayed with visible data rows.		UI	Not Executed	Dilrangi 214102J	
UI/TC10	Verify "Add Plant" button visibility	Validates that the "Add Plant" functionality is strictly hidden from standard users.	1. The user needs to log in as a Standard User (Non-Admin).  2. The user has navigated to the Plant List page (/ui/plants).	1. Observe the top section (header/toolbar) of the Plant List page.  2. Check for the presence of the "Add Plant" button.	• The "Add Plant" button is NOT visible on the screen.		UI	Not Executed	Dilrangi 214102J	
UI/TC11	Verify Plant List Pagination user as non admin	It ensures that when the number of plants exceeds the default view limit (e.g., 10), a standard user can navigate to subsequent pages.	1. The user needs to log in as a Standard User (Non-Admin).  2. The user is on the Plant List page.  3. More than 10 plants exist in the database.	1. Scroll to the bottom of the plant list table.  2. Click on the "Next" button or a specific page number (e.g., "2")	• The plant list updates to display the next set of records.  • The active page indicator highlights the current page number.		UI	Not Executed	Dilrangi 214102J	
UI/TC12	Verify "Sorting by Name", "Sorting by Price", "Sorting by Stock quantity" in the Plant List	Validates the multi-column sorting capabilities of the Plant List. It ensures that a standard User can organize the inventory view by clicking on "Name", "Price", or "Stock Quantity" headers to toggle between ascending and descending order.	1. The user needs to log in as a Standard User (Non-Admin).  2. The user is on the Plant List page (/ui/plants).  3. Multiple plants exist with varying names, prices, and stock quantities.	1. Click the "Name" column header once, then click it again.  2. Click the "Price" column header once, then click it again.  3. Click the "Stock Quantity" column header once, then click it again.	• Name: Sorts alphabetically Ascending (A-Z), then Descending (Z-A).  • Price: Sorts by value Low-to-High, then High-to-Low.  • Stock Quantity: Sorts by count Low-to-High, then High-to-Low.		UI	Not Executed	Dilrangi 214102J	
UI/TC13	Verify Filter by Plant Category as an User(non admin)	Validates the search and filtering capabilities for standard users. The user can isolate specific types of plants by selecting a category, displaying only the relevant items in the table.	1. The user needs to log in as a Standard User (Non-Admin).  2. The user is on the Plant List page.  3. Plants with assigned categories exist in the system.	1. Locate the Category dropdown menu.  2. Select an existing category (e.g., "Indoor").  3. Click the "Search" or "Filter" button.	• The table refreshes to display only the plants matching the selected category.  • Non-matching plants are hidden from the view.		UI	Not Executed	Dilrangi 214102J	

API/TC14	Verify that admin can retrieve the plant list	This API test verifies that an authenticated admin user can retrieve a paginated list of plants from the system.	1. Admin user account exists and is active. 2. Admin user is authenticated and has a valid access token. 3. At least one plant record exists in the system.	1. Authenticate as an Admin user and obtain a valid access token. 2. Send a GET request to the Get Plant List API endpoint ( <a href="http://localhost:8080/api/plants">http://localhost:8080/api/plants</a> ). 3. Observe the API response.	<ul style="list-style-type: none"> <li>The API returns HTTP 200 OK.</li> <li>The response contains a paginated list of plant records with correct details.</li> </ul>		API	Not Executed	Dilrangi 214102J	
API/TC15	Verify admin can add a valid plant	This API test verifies that an authenticated admin can successfully add a plant using valid input data.	1. Admin user account exists and is active. 2. Admin user is authenticated and has a valid access token.	1. Authenticate as an Admin user. 2. Send a POST request to " <a href="http://localhost:8080/api/plants/category/{id}">http://localhost:8080/api/plants/category/{id}</a> " with valid payload 3. Observe the API response.	<ul style="list-style-type: none"> <li>The API returns HTTP 201 Created.</li> <li>The response contains the newly created plant details.</li> <li>The plant is persisted in the database.</li> </ul>		API	Not Executed	Dilrangi 214102J	
API/TC16	Verify validation errors when adding an invalid plant when admin add new plant	This API test verifies that the system enforces validation rules when invalid plant data is submitted.	1. Admin user account exists and is active. 2. Admin user is authenticated and has a valid access token.	1. Authenticate as an Admin user. 2. Send a POST request to ( <a href="POST /api/plants">POST /api/plants</a> ) with invalid payload: ex: price in negative, and quantity in negative 3. Observe the API response.	<ul style="list-style-type: none"> <li>The API returns HTTP 400 Bad Request.</li> <li>Error messages are returned:            "Price must be greater than 0"            "Quantity cannot be negative"         </li> </ul>		API	Not Executed	Dilrangi 214102J	
API/TC17	Verify plant list pagination API	This API test verifies that plant records can be retrieved using pagination parameters.	1. Admin user account exists and is active. 2. Admin user is authenticated and has a valid access token. 3. More than 10 plant records exist in the database.	1. Authenticate as an Admin user. 2. Send a GET request to <a href="http://localhost:8080/api/plants/paged?page=0&amp;size=10&amp;sort=name,asc">http://localhost:8080/api/plants/paged?page=0&amp;size=10&amp;sort=name,asc</a> 3. Observe the API response.	<ul style="list-style-type: none"> <li>The API returns HTTP 200 OK.</li> <li>The response contains the first page of plant records.</li> <li>Pagination metadata is correct.</li> </ul>		API	Not Executed	Dilrangi 214102J	
API/TC18	Verify filter plants by category	This API test verifies that an admin can filter plant records by category.	1. Admin user account exists and is active. 2. Admin user is authenticated and has a valid access token. 3. Plants with different categories exist in the system.	1. Authenticate as an Admin user. 2. Send a GET request to " <a href="http://localhost:8080/api/plants/category/{id}">http://localhost:8080/api/plants/category/{id}</a> " 3. Observe the API response.	<ul style="list-style-type: none"> <li>The API returns HTTP 200 OK.</li> <li>Only plants belonging to the selected category are returned.</li> <li>Plants from other categories are excluded.</li> </ul>		API	Not Executed	Dilrangi 214102J	
API/TC19	Verify search plant by name by user as a non admin	This API test verifies that a non-admin user can successfully search for plants by a name query parameter.	1. User account exists and is active. 2. User is authenticated with a valid access token. 3. Plant record "Rose" exists in system.	1. Authenticate as a User (non-admin). 2. Send a GET request to ' <a href="http://localhost:8080/api/plants/paged?name=Rose&amp;page=1&amp;size=10&amp;sort=name,asc">http://localhost:8080/api/plants/paged?name=Rose&amp;page=1&amp;size=10&amp;sort=name,asc</a> ' 3. Observe response.	<ul style="list-style-type: none"> <li>The API returns a successful response (HTTP 200).</li> <li>The response body contains an array of plant objects where the name matches the query</li> </ul>		API	Not Executed	Dilrangi 214102J	

API/TC20	Verify User access to Admin-only Delete endpoint	This API test ensures that a non-admin user is restricted from accessing Delete backend actions.	1. User account exists and is active. 2. User is authenticated (non-admin).	1. Send a DELETE request to /api/plants/{id} using the User's token.	<ul style="list-style-type: none"> <li>The API returns HTTP 403 Forbidden for non-admin tokens.</li> <li>The unauthorized request to Delete/Edit is blocked at the server level.</li> </ul>		API	Not Executed	Dilrangi 214102J	
API/TC21	Verify User access to Admin-only update endpoint	This API test ensures that a non-admin user is restricted from accessing Edit backend actions.	1. User account exists and is active. 2. User is authenticated (non-admin).	1. Send a PUT request to /api/plants/{id}.	<ul style="list-style-type: none"> <li>The API returns HTTP 403 Forbidden for non-admin tokens.</li> <li>The unauthorized request to Delete/Edit is blocked at the server level.</li> </ul>		API	Not Executed	Dilrangi 214102J	
API/TC22	Verify "No plants found" search and filter result	This API test verifies that the system returns an empty list when search or filter criteria do not match any records.	1. Authenticated as Admin or Non admin. 2. No plants exist with the name "Cactus" or category "Desert". 3. admin or non admin account exists and active	1. Send a GET request to "http://localhost:8080/api/plants/page?name=AllTestsrt&categoryId=10&page=1&size=10&sort=name,asc" 2. Observe the response list.	<ul style="list-style-type: none"> <li>API returns HTTP 200 OK.</li> <li>Response body is an empty array [] (allowing the UI to display the "No plants found" message).</li> </ul>		API	Not Executed	Dilrangi 214102J	
API/TC23	Verify get plant by invalid id	Verify that the system handles requests for non-existent plant IDs gracefully and returns a proper error response.	1. Authenticated as a non admin. 2. Call GET plant API with invalid ID. 3. non admin account exists and active	1. Send a GET request to /api/plants/{id}.	<ul style="list-style-type: none"> <li>API returns HTTP 404 OK.</li> <li>Error response indicating plant not found.</li> </ul>		API	Not Executed	Dilrangi 214102J	