

## ● Login page

TC ID	Test Objective	Technique	Explanation of the Technique
FR-1_TC1	Verify that valid users can log in and invalid credentials are blocked.	Positive & Negative Testing	Checks correct system behavior with valid inputs (positive) and proper error handling with invalid inputs (negative).
FR-1_TC2	Verify that the system logs out the user and redirects to the login page.	Positive Testing / Use Case Testing	Validates normal expected behavior following a basic user flow.
FR-1_TC3	Ensure "Forgot your password?" The link is visible and functional.	UI Testing	Confirms presence, visibility, and working navigation of a user interface element.
FR-1_TC4	Verify that users can change their password successfully.	Positive Testing / Functional Testing	Confirms correct system behavior using valid data.
FR-1_TC5	Verify system validation messages for incorrect or mismatched passwords.	Negative Testing	Focuses on error handling for invalid user input.
FR-2_TC1	Ensure users can successfully book a free demo and receive confirmation email.	Positive Testing / End-to-End Testing	Tests a full valid flow from submitting the form to receiving the email.
FR-2_TC2	Verify that users can contact sales and receive one confirmation email.	Functional Testing	Ensures the feature performs expected operations and avoids defects (such as duplicate emails).
FR-2_TC3	Verify that users can only select future dates when booking a meeting.	Boundary Value Analysis	sets restrictions around date selection—past vs future values.

FR-2_TC4	Ensure users can register for a free trial and receive only the relevant email.	Functional Testing	Checks system logic consistency and avoids unintended email triggers.
FR-2_TC5	Verify proper handling of multiple activation link clicks.	Negative Testing / Error Handling Testing	Validates that the system prevents repeated activation attempts.
FR-2_TC6	Verify successful login using credentials from the activation email.	Positive Testing	Ensures the system works properly with valid, system-generated data.
FR-2_TC7	Confirm that the system sends a welcome email with login details post-activation.	Functional Testing	Ensures correct system-generated communication after workflow completion.
FR-2_TC8	Verify the dropdown for "Number of Employees" works and allows selection.	UI Field Validation Testing	Ensures a form field is functional, responsive, and behaves correctly.
FR-2_TC9	Verify the phone number field supports valid numbers including country codes.	Input Validation Testing	Ensures the input field meets required formatting and validation rules.

## ● Recruitment

TC ID	Test Objective	Technique	Explanation of the Technique
TC_HR_001	Verify HR can <b>access the job</b> posting page.	<b>Use Case Testing</b> (Successful Path)	Validates the first step of the core recruitment scenario.
TC_HR_002	Verify HR can <b>post a new job</b> vacancy.	<b>Equivalence Partitioning</b> (Valid Class)	Tests the "Happy Path" by confirming success with valid and complete data.

TC_HR_003	The Verify system <b>validates required job fields</b> .	<b>Boundary Value Analysis</b> (Minimum Boundary)	Specifically tests the lower boundary of required input (Null/Empty values) to check validation messages.
TC_HR_004	Verify job vacancy is <b>displayed to candidates</b> .	<b>End-to-End Testing</b>	Checks the entire workflow result, from HR posting to the Candidate view on the public side.
TC_HR_005	Verify HR can <b>track applications</b> (viewing data).	<b>Use Case Testing</b>	Validates the core HR function of viewing applicant data.
TC_HR_006, 007	Verify <b>Search and Filter by Job Title</b> functionality.	<b>Equivalence Partitioning</b> (Specific Filter Value)	Uses specific, valid filter values (Job Titles, Keywords) to partition and test large data sets.
TC_HR_008	Verify <b>input of invalid data</b> in Job Title.	<b>Negative Testing</b> (Error Guessing)	Tests system robustness against non-standard or malicious input (e.g., special characters, scripts).
TC_HR_009	The Verify system <b>prevents duplicate entries</b> .	<b>Negative Testing</b> (Business Rule Constraint)	Checks a critical business rule to ensure data uniqueness is enforced.
TC_HR_010	Verify HR can <b>delete</b> a posted job vacancy.	<b>Use Case Testing</b> (CRUD Operation)	Validates the successful execution of the Delete function.
TC_HR_011	Verify <b>email/notification</b> after posting a job.	<b>Non-Functional Testing</b> (Integration)	Checks the system's integration with the email or notification service.

<b>TC_HR_012</b>	Verify job posting page <b>responsiveness</b> .	<b>Non-Functional Testing</b> (Compatibility/ Usability)	Checks the adaptability and appearance of the UI across different screen sizes (mobile, desktop).
<b>TC_HR_013</b>	Verify <b>permission control</b> for HR (Non-HR access).	<b>Role-Based Access Control (RBAC)</b>	Explicitly checks the security requirement that unauthorized users are restricted from accessing critical functions.
<b>TC_HR_014</b>	Verify candidate application <b>status update</b> .	<b>State Transition Testing</b>	Validates that the system correctly manages the change of state (e.g., Shortlisted $\rightarrow$ Rejected) according to the expected workflow.

- User Management

TC_ID	TC objective	Technique	Explanation of technique
<b>TC_UserRoleManage_001</b>	Check the admin role has permissions to full access modules	Equivalence partitioning	Admin role represents one valid equivalence class — verifying that users in this class have full access
<b>TC_UserRoleManage_002</b>	Check the ESS role has permissions to access to personal info and leave requests only	Equivalence partitioning	ESS role is equivalence class with limited access — ensures correct behavior for this category

<b>TC_UserRoleManage_003</b>	Check the manager role has permissions to access to team data, leave approvals, and .reports only	Equivalence partitioning	The manager role is a different equivalence class — checks access is limited to specific .modules
<b>TC_UserRoleManage_004</b>	Validate that the user should only have one role	Boundary value analysis	Tests the boundary between “one role” (valid) and “more .(than one role” (invalid
<b>TC_UserRoleManage_005</b>	Ensure system allow to assign ESS role and allow to update to another role	Equivalence partitioning	Assigning a valid ESS role and updating to another valid role — tests valid partitions of .allowed role changes
<b>TC_UserRoleManage_006</b>	Ensure system allow to assign admin role and allow to update to another role	Equivalence partitioning	Assigning a valid Admin role and updating to another valid role — tests valid partitions of .allowed role changes
<b>TC_UserRoleManage_007</b>	Ensure system allow to assign manager role and allow to update to another role	Equivalence partitioning	Assigning a valid manager role and updating to another valid role — tests valid partitions of .allowed role changes
<b>TC_UserRoleManage_008</b>	Ensure that system allow to revoke roles	State transition	Tests transition from Active → Revoked state; ensures user .cannot log in after revocation
<b>TC_UserRoleManage_009</b>	Verify that when a user role is assigned, the action is logged in the .system audit	State transition	Verifies the state change User Created → Role Assigned is logged properly in the audit .system
<b>TC_UserRoleManage_010</b>	Verify that when a user role is updated the action is logged .in the system audit	State transition	Verifies transition Role Updated .→ Audit Log Recorded

TC_UserRoleManage_011	Verify that when a user revoked the action is logged in the system audit	State transition	Verifies transition User Revoked → Log Entry .Recorded
TC_UserRoleManage_012	Verify that every user should have a role	Negative Testing	Ensures the system handles invalid actions or inputs properly

## إدارة الإجازات | FR-4: Leave Management

TC_ID	TC Objective	Technique	Explanation of Technique
TC_LM_001– Leave List	Verify manager can view leave requests and filters work	Equivalence Partitioning + Functional Testing	Inputs (filters) are divided into valid/invalid groups and tested to ensure correct output.
TC_LM_002– Assign Leave	Verify manager/admin can assign leave	Positive & Negative Testing	Ensures the system behaves correctly with valid data and handles missing/invalid entries.
TC_LM_003– Bulk Assign Leave	Validate leave assignment submission	Boundary Value Analysis	Tests date range boundaries and leave balance calculations..
TC_LM_004– Apply Leave	Verify employee can apply for leave	Use Case Testing	Follows real user flow from login → apply leave.
TC_LM_005– My Leave Usage	Verify correctness of leave usage summary	Equivalence Partitioning	Validated displayed data belongs to expected partitions (entitlement, taken, balance).
TC_LM_006– Leave Calendar	Verify leave calendar displays correct results	Functional Testing	Tests display and search functionality.
TC_LM_007– My Leave	Validate employee leave history	Use Case Testing	Follows actual end-to-end employee actions

TC_LM_008– Entitlements	Verify admin can view and manage entitlements	Functional + Regression Testing	Ensures entitlement list loads .and updates correctly
TC_LM_009– Employee Entitlements Calculation	Verify entitlement totals	Boundary Value + Calculation Verification	Confirms accuracy of totals and leave days.
TC_LM_010– Add New Entitlement	Validate admin can create new entitlement	Positive Testing	Ensuring correct data leads to successful saves.
TC_LM_011– Leave Period Lock	Validate leave period cannot be edited	Negative Testing	Ensures the system prevents modifications to fixed fields.
TC_LM_012– Leave Types Display	Validate Leave Types list	functional Testing	checks list integrity and attribute correctness.
TC_LM_013– Add Leave Type	Ensure new leave type can be added	Positive Testing	checking correct form input leads to creation.
TC_LM_014	Validate required fields	Negative Testing	Ensures missing fields show errors.
TC_LM_015	Validate system allows same name for different countries	Equivalence Partitioning	Compares behavior between same-name/same-country vs same-name/different-country.
TC_LM_016– Holidays	Ensure admin can set official holidays	Functional Testing	Validates addition and saving of holiday details.
TC_LM_017 Multi-Country Holiday Selection (Bug-001)	Validate holiday can have multiple countries	Negative Testing (Bug Confirmation)	confirms defects by verifying only one country can be selected.
TC_LM_018– Working Weekends Leave Types	Validate leave types list	Functional Testing	Ensures predefined leave types match configurations.

TC_LM_019– Multi-Country Holiday Selection (Bug-002)	Validate selection of multiple locations	Negative Testing (Bug Reproduction)	Verifies the system’s incorrect restriction.
TC_LM_020– Bradford Factor Threshold	Validate threshold values displayed	Functional Testing + Boundary Analysis	ensures numeric thresholds match expected boundary values.
TC_LM_021– Notifications	Validate admin can create leave notifications	Use Case Testing	Tests full flow from creation to list display.
TC_LM_022 – Leave Calendar Configuration	Validate calendar filters function correctly	Exploratory + Functional Testing	Mixes functional checks with exploratory testing because filters repeat (Bug-003).

الأمان | **NFR-2: Security**

TC_ID	TC Objective	Technique	Explanation of Technique
NFR-2_TC1	Verify passwords are stored as hashed values	Security Testing	Ensures sensitive data (passwords) is not stored in plain text — validates database .security
NFR-2_TC2	Verify hashing algorithm type (e.g., SHA-256 or encrypt	Configuration Testing	Confirms only approved secure algorithms are used in system .configuration
NFR-2_TC3	Verify secure transmission of passwords over HTTPS	Penetration Testing	Checks that passwords are transmitted securely (no (exposure in network traffic



- Employee Information

TC ID	TC Objective	Technique	Explanation of Technique
TC_001	Verify Admin can open Add Employee form 11111	Positive Testing / Functional Testing	Ensures the basic function (opening the form) works successfully when performed by an authorized role (Admin) <sup>22222</sup> .
TC_002	Verify mandatory fields (Employee ID, First Name, Last Name) are required 33333	Negative Testing / Boundary Value Analysis (BVA)	Confirms the system rejects the transaction when required fields are left blank, testing the lower boundary condition of zero input <sup>44444</sup> .
TC_003	Verify record can be saved with valid details 55555	Equivalence Partitioning / Positive Testing	Uses a set of valid input data from a single equivalence class to ensure the save operation succeeds <sup>66666</sup> .
TC_004	Verify Employee ID must be unique 777777	Negative Testing / Validation Testing	Attempts to input a duplicate Employee ID to ensure the system rejects it and displays the "Employee id already exists" error message <sup>888888888</sup> .
TC_005	Verify Employee ID cannot be empty 99999	Negative Testing / Validation Testing	Confirms the system displays an error when the mandatory Employee ID field is left blank <sup>1010101010</sup> . <b>(Note: Actual Result showed a Fail)</b> <sup>111111</sup> .

TC_006	Verify Employee ID format rule enforced 1212121212	Validation Testing	Ensures the system enforces and validates the correct format rules (e.g., numeric or alphanumeric) for the Employee ID 1313131313
TC_009	Verify unauthorized user cannot access Add form 141414141414	Security Testing / Role-Based Access	Attempts to access the "Add Employee" function using roles without permission (Manager/ESS) to ensure access is denied 1515151515151515
TC_012	Verify Employee ID field is non-editable once saved 1616161616	Validation Testing	Attempts to modify the Employee ID after the record is created to ensure the field maintains its non-editable property 1717171717 . <b>(Note: Actual Result showed a Fail)</b> 181818
TC_016	Verify invalid data (email, phone) cannot be saved 191919191919	Validation Testing	Inputs improperly formatted data (e.g., invalid email) to ensure the system rejects it and provides expected format messages 2020202020202020
TC_017	Verify Manager limited update access 212121212121	Role-Based Access / Security Testing	Confirms the Manager role can only update fields within their permitted scope, while restricted fields are disabled or not visible 222222222222222222

TC_023	Verify partial name search works 232323232323	Functional Testing / Search Testing	Inputs a partial name (e.g., "Ali") to ensure the search returns relevant matching records 2424242424242424
TC_027	Verify search performance (response time) 252525252525	Performance Testing	Measures the time taken to load search results to ensure they appear within acceptable response time limits (e.g., < 3 seconds) 2626262626262626
TC_028	Verify names reject special characters 2727272727	Validation Testing	Attempts to input numbers or special characters into the name field to ensure the system rejects them 2828282828 <b>(Note: Actual Result showed a Fail)</b> 292929
TC_029	Verify max character length respected 303030303030	Boundary Testing	Inputs a value exceeding the maximum character length defined in the SRS to ensure the constraint is enforced 3131313131313131
TC_031	Verify trimming of extra spaces 3232323232	Validation Testing	Inputs text with unnecessary leading/trailing spaces to ensure the system automatically removes (trims) them 3333333333 <b>(Note: Actual Result showed a Fail)</b> 343434

TC_033	Verify all updates logged with timestamp/user/action 3535353535	Audit Testing	Confirms the audit log contains all required update details (timestamp, user ID, action performed) for compliance and traceability 3636363636. (Note: Actual Result was Blocked/Not Applicable) 373737.
TC_035	Verify session timeout restricts access 383838383838	Security Testing	Validates that user access is revoked after a session timeout until the user logs in again 3939393939393939.