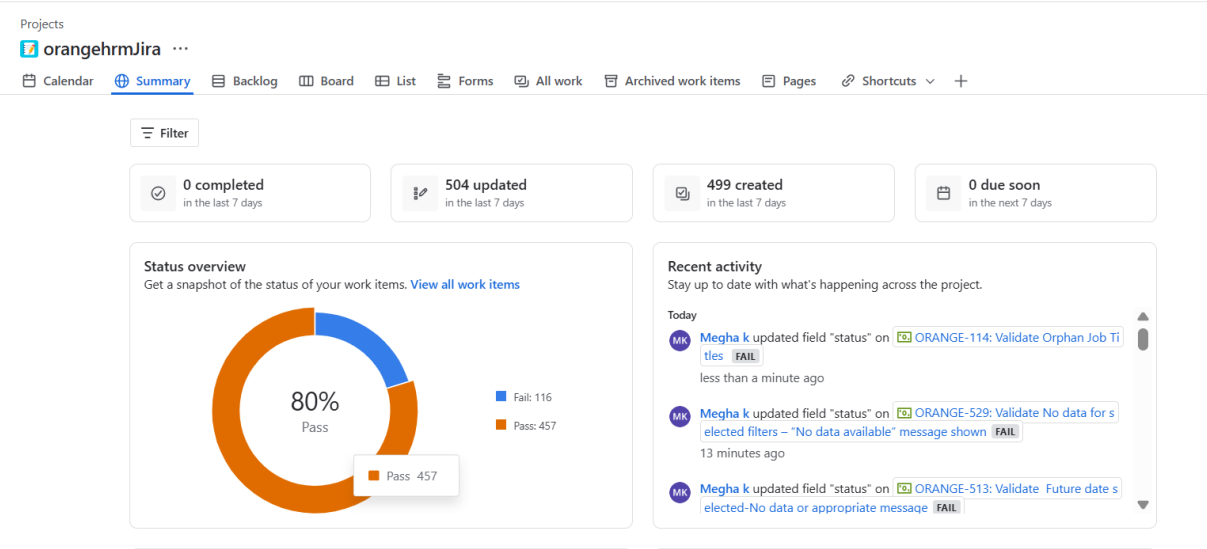
**Bug Report for OrangeHRM**

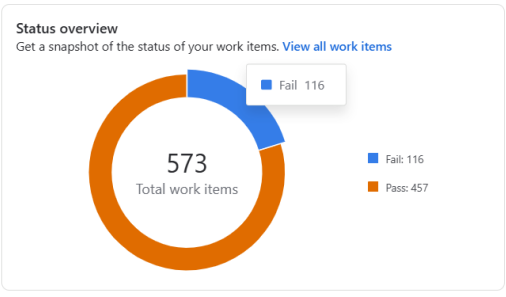
Megha K

June 15, 2025

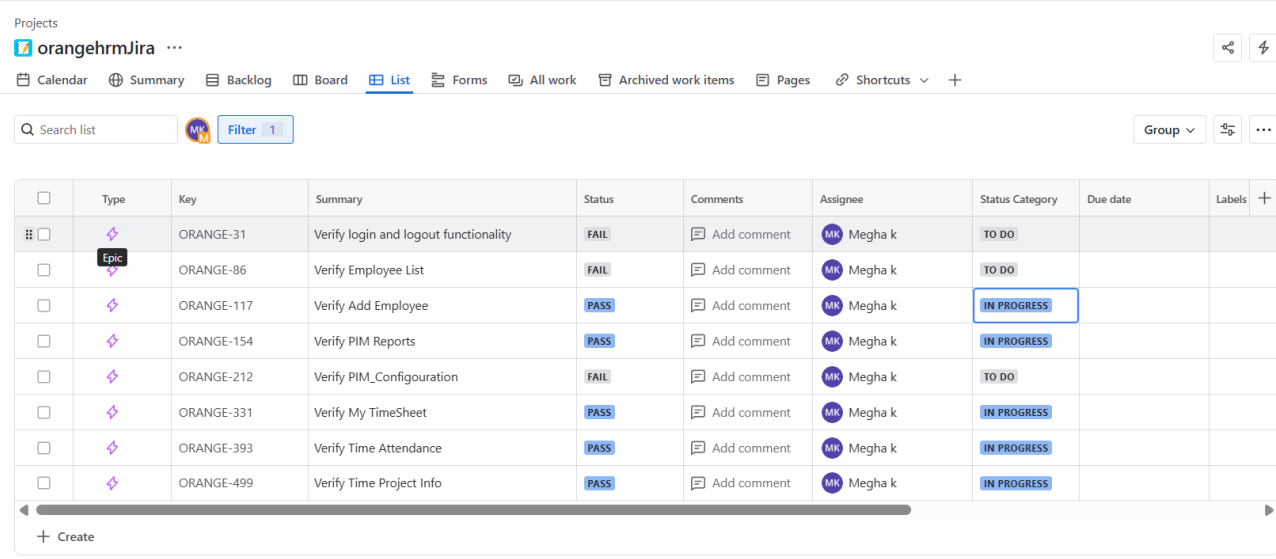
**Summary**



Total Work items :-



Epic :-



Priority Breakdown:-



**Bugs:-**

1. Scenario :- Validate login with one user role , try accessing URL of another role that a user with one role cannot access pages or URLs assigned to another role in OrangeHRM.

Bug Description :- ESS user was able to access Admin-only pages and view user management content. Major security issue.

Type of Testing :- Component Integration Testing

Test Technique - Negative

Test method : Manual

Test steps:

1. Open the browser and navigate to the OrangeHRM login page

2. Login using ESS (Employee Self service ) User credentials

3. Confirm successful login and visibility of limited ESS features (Eg , My Info, Leave)

4. In the same session manually enter direct URL,that is restricted to admin user ,(User management page)

5. Press enter and observe system response .

6. Attempt similar access to other restricted modules like job titles

Pre condition:-

1.OrangeHRM is installed or accessible (e.g., https://opensource-demo.orangehrmlive.com/).

2.Two or more user roles exist (e.g. Admin, ESS (Employee Self-Service), Manager).

3.You have login credentials for a lower-privilege role (e.g., ESS user).

4.You know or can obtain the URL of a page restricted to a higher-privilege role (e.g., Admin Dashboard or User Management).

Test data :-

1.ESS username :- ess\_user(create in admin panelif testing locally )

2.Role :- Employee Self service

Severity :-High : Potential security vulnerability if users can access unauthorized resources

Priority :- High :- critical for maintaining role based access restrictions in HR system.

Expected Result :-

1.The user should be denied access to Admin-only pages.

2.The system should:

3.Redirect the user to a dashboard or error page, or

4.Show an "Access Denied" or "You are not authorized" message.

5.No restricted data or functionality should be visible to the unauthorized user.

Actual result :-

1.If access control works correctly:

2.ESS user was blocked from accessing the Admin page URL. OrangeHRM redirected to the dashboard with an “Access Denied” message.

3.If access control fails:

ESS user was able to access Admin-only pages and view user management content. Major security issue.

4.If partial access occurs:

Page loaded partially (UI elements visible), but actions failed with permission errors. Indicates front-end filtering but weak back-end checks.

**--------------------------------------------------------------------------------------------------------------------------------------**

2.Scenario :- Validate absence of loading spinner or feedback on login delay in OrangeHRM

note :- the login page displays a loading spinner or some form of feedback when login is delayed (e.g., due to slow network or server response).

Testing method :- Manual /Exploratory Testing

Testing technique :- black box testing

Type of Testing :- Functional UI Testing

Test steps :-

1.Open the browser and go to OrangeHRM login page

2. Open browser dev tools and set

3.Enter valid credentials

4.Click Login

5. Observe if there is any loading spinner , Progress bar ,or visual indication that login is in progress

6. Wait for dash board to load .

Pre condition:-

1.OrangeHRM is accessible via a browser (e.g., https://opensource-demo.orangehrmlive.com/).

2.User has valid login credentials.

3network conditions can be throttled to simulate delay (using chrome ->DevTools ->Network ->Slow 3G or Offline/Online toggle )

Test data :-

Username : Admin

Password : admin123

Network : simulated slow connection

Severity :- Medium :Does not impact security or function , affects user experience especially on slow connection.

Priority :- Medium:Should be addresed to improve usability , particularly users in low bandwidth environments.

Expected Result :-

1.After clicking login button application should show loading spinner , animation, logging in " message to indicate request is being processed

2.User should not be felt uncertain about whether the action is Working

Actual Result :-

Failed :-

1. When loggin is delayed(due to slow network , no spinner or loading indicator appears .
2. login button remains clickable and user recieves no visual feedback that the system is processing the request .This can cause confusion or lead the user to click multiple times.
3. **Bug Description :-** ESS user was blocked from accessing the Admin page URL. OrangeHRM redirected to the dashboard with an “Access Denied” message.

# Validate login with one user role , try accessing URL of another role

user with one role cannot access pages or URLs assigned to another role in OrangeHRM.

Type of Testing :- Component Integration Testing  
Test Technique - Negative  
Test method : Manual

Steps:-

1. Open the browser and navigate to the OrangeHRM login page
2. Login using ESS (Employee Self service ) User credentials
3. Confirm successful login and visibility of limited ESS features (Eg , My Info, Leave)
4. In the same session manually enter direct URL,that is restricted to admin user ,(User management page)
5. Press enter and observe system response .
6. Attempt similar access to other restricted modules like job titles

Pre condition:-

1.OrangeHRM is installed or accessible (e.g., <https://opensource-demo.orangehrmlive.com/).>

2.Two or more user roles exist (e.g. Admin, ESS (Employee Self-Service), Manager).

3.You have login credentials for a lower-privilege role (e.g., ESS user).

4.You know or can obtain the URL of a page restricted to a higher-privilege role (e.g., Admin Dashboard or User Management).

Test data :-

ESS username :- ess\_user(create in admin panelif testing locally )  
Role :- Employee Self service

Severity :-

High : Potential security vulnerability if users can access unauthorized resources

Priority :-

High :- critical for maintaining role based access restrictions in HR system.

Expected Result :-The user should be denied access to Admin-only pages.

The system should:

1.Redirect the user to a dashboard or error page, or

2.Show an "Access Denied" or "You are not authorized" message.

3.No restricted data or functionality should be visible to the unauthorized user.

Actual result :-

1. If access control works correctly:

ESS user was blocked from accessing the Admin page URL. OrangeHRM redirected to the dashboard with an “Access Denied” message.

2.If access control fails:

ESS user was able to access Admin-only pages and view user management content. Major security issue.

3.If partial access occurs:

Page loaded partially (UI elements visible), but actions failed with permission errors. Indicates front-end filtering but weak back-end checks.

--------------------------------------------------------------------------------------------------------------------------------------

1. **Validate user logs in to the application using standard valid Username and Password.]**

Login page reloads or hangs  
Error message is shown despite valid credentials  
Dashboard does not load

**Bug Description:-** Login page reloads

Test Techniques :-  
1.Equivalence Partitioning:-  
Valid partition: username/password that meet all criteria (e.g., correct format, length, exists in DB).  
Invalid partition: incorrect username/password, empty fields, special characters inappropriately used, etc.

2.Boundary Value Analysis  
Username/password at minimum and maximum allowable lengths.  
E.g., password of exactly 8 and 16 characters (if limits are 8–16 chars).

1)Positive Testing  
Provide valid username and password → Expect success.

2)Negative Testing (for comparison or robustness)  
Input invalid combinations → Expect failure (e.g., wrong password, blank fields).

3.State Transition Testing  
Test transitions from "logged out" to "logged in" state when valid credentials are entered.

Test Method :-

1.Manual Testing  
Manually enter credentials into UI and verify login success.

Test Type :-  
1.Functional Testing: Verifies that login works as intended.  
2.Regression Testing:Ensure login continues to work after updates.

Steps :-1. Launch the Browser 2. Enter the login URL 3. Enter Valid Username and Password 4. Click on login button.

Pre condition:-

1.The user is on the login page

2.Username and password are valid and active in the system

Test data :-Username : Admin Password : admin123

Severity :high

Priority :high

Expected Result :-

1.The user is successfully authenticated

2.The user is redirected to the OrangeHRM Dashboard (home page)

[3.No](http://3.No) error messages are displayed

1. The dashboard contains welcome text (e.g., "Welcome Admin") and navigation options

Actual result :-

Pass :- The user is redirected to the Dashboard/Home page  
Correct user name is shown  
System behaves as expected  
last\_login is updated (note )

Fail :- Login page reloads or hangs  
Error message is shown despite valid credentials  
Dashboard does not load

**--------------------------------------------------------------------------------------------------------------------------------------**

5. Validate the visual feedback on incorrect login (red border /alerts)

Bug description:- No error message is displayed message is confusing /vague.User may be redirected to left unsure of login failed.

Type of Testing :- UI , functional

Test technique :- Black box testing

testing method : manual Testing

Steps:-

1. Open the OrangeHRM login Page

2. Enter invalid login credentials

3. Click Login

4. Observe visual feedback (red border)

Pre condition :-

Login page loaded

Test data :-

Invalid username/password

Severity :- medium:- Impacts user experience and clarity espcially for non- technical users

Priority :- High :- clear feedback on login errors is essential for usability and reducing support queries .

Expected Result :- System should display

1. Clear error message

such as invalid credentials

2. Input fields should shake/vibrate or highlight in red to indicate error visually

3. Prevent login and keep the user in Login page.

4. no redirect to dashboard or any protected area.

Actual result :-

Pass : After entering incorrect credential appears in red below the login form . Fields remain filled or reset . No login occurs and users stay in login screen

Fail : No error message is displayed message is confusing /vague.User may be redirected to left unsure of login failed.

--------------------------------------------------------------------------------------------------------------------------------------

1. Validate login failure when using an invalid username with a valid password

Bug Description :- No Error message shown "it says internal server error "

Testing Technique:- Black box testing

Type of testing :- Functional ,Security,negative

Method : - manual

Steps :-

1. Launch the browser and navigate to the OrangeHRM login page

2. Enter the URL

3. Enter invalid username and valid password

4.Click on the Login button

Pre condition:-

1.The user is on the login page

2.Username and password are valid and active in the system

Test data :-

Username : Admin\*\*\*

Password : admin123

Severity :High

Priority :high

Expected result :- Should show valid Username or Password

Actual result :- No Error message shown "it says internal server error "

--------------------------------------------------------------------------------------------------------------------------------------

7.Validate that login is denied when using disabled or inactive user credentials.

Bug Description :- .There is no indication that the account is inactive.

Types of Testing : Security , Functional ,Negative

test techniques :- Black box testing

Methods :- Manual Testing

Steps:-

1. Launch the browser

2.Open OrangeHRM login page

3. Login using admin credential

4. Make user Inactive

Admin-> Usermanagement -> Users

5. Search for active user account

6. Click edit on User

7.Change status for enabled to disabled

8. Save the Changes

9.Attempt login using inactive user

10.observe result

Pre condition:- User account john.doe is marked as inactive or disabled in the system

Test data :-

Username: john.doe

Password: ValidPass@123

Severity :high

Priority :High

Expected Result :-

Login fails.

- Show appropriate message like "Account is inactive", "User disabled", or generic "Invalid credentials" (if security policy avoids account status disclosure).

- No access is granted.

Actual Result :-

Fail :-

1.The system shows a generic error like:

“Invalid username or password.”

2.There is no indication that the account is inactive.

3.In some cases, the system may allow login or partial access (a serious issue).

4.Login proceeds (critical security bug)

Pass:-

1.Popup Error message should be displayed , Login should fail.

"Invalid credentials".

2.Login is blocked and proper error shown (correct behavior)

--------------------------------------------------------------------------------------------------------------------------------------

8.Validate session behavior when logging into OrangeHRM in multiple browser tab using same user .

Bug Description:- One tab logs out the other

Test type :- Functional, Session management ,

Test Technique : - State transition Testing ,Exploratory Testing for edge cases.

Test Method :-manual Testing

Steps:

1.Open Tab 1 in Chrome or Firefox.

2.Go to the OrangeHRM login page and log in as Admin.

3.Open Tab 2 in the same browser session.

4.Navigate to the OrangeHRM login page in Tab 2.

5.Log in again using the same credentials (Admin).

6.Perform the following checks:

7.Refresh Tab 1 and observe if still logged in.

8.Perform actions in both tabs (e.g., view dashboard, open "My Info").

9.Optionally, logout from one tab and observe behavior in the other.

Pre condition:-

User Admin exists and is active.

Browser allows multiple tabs.

Session timeout is set (e.g., 20–30 mins).

JavaScript and cookies are enabled.

User data :-

Username : Admin

Password : admin123

Severity :medium

Priority :medium

Expected result :-

1.Login is successful in both tabs.

2.Session remains valid in both tabs.

3.Actions in one tab do not log out or conflict with the other.

4.Logging out in one tab ends session in both tabs (if session is shared).

Actual Result :-

Pass – If session remains consistent and expected actions work in both tabs.

Fail – If:

One tab logs out the other

Session corruption occurs

Errors occur when performing actions in parallel.

--------------------------------------------------------------------------------------------------------------------------------------

9.Validate clicking the logout button ends the session and redirects the user to the login page.

Bug Description:- Session remains active

Functional Testing :Ensures logout button performs expected behavior

Session Management Testing :Validates session termination upon logout

Test Technique :-

1.Scenario-Based Testing :Simulate user logout from a valid session

2.State Transition Testing :From “logged in” → “logged out” state

3.Negative Testing :Try accessing dashboard after logout via browser history or URL

Test Method :-

Manual Testing :Performed through the UI and browser behavior

Steps:-

1.Log in to OrangeHRM with:

Username: Admin

Password: admin123

2.Verify you are redirected to the dashboard/home page.

3.Locate and click the Logout button (usually in the top-right corner or in the user menu).

4.Observe the behavior after clicking.

Pre condition:-

1.The user Admin exists and has a valid login.

2.User is already logged in and is on the OrangeHRM Dashboard.

3.Browser allows session cookies and JavaScript execution.

Test data :-

Username : Admin

Password : admin123

Severity : High – Broken logout can cause session hijacking risk.

Priority:- High – Logout must work reliably for compliance and security.

Expected Result :-

1.The session is terminated.

2.User is redirected to the login page.

3.URL should now show /auth/login or similar.

4.Session cookies should be cleared or expired.

5.User cannot go back to the dashboard using the browser's back button.

Actual Result :-

1.Pass – If user is fully logged out, redirected, and session is cleared.

2.Fail – If:

User is not redirected

Session remains active

User can still access protected pages after logout

“Logout” link doesn’t work or throws an error

--------------------------------------------------------------------------------------------------------------------------------------

# Validate login with the same credentials from multiple devices or browsers at the same time /Validate application behavior when logging in simultaneously from multiple browsers or devices using the same credentials.

Bug Description :- User logs in on second browser, and first session is not terminated, but starts malfunctioning or silently stops receiving updates.

Test Type :-  
1.Session Management Testing :Evaluates how sessions behave across devices/browsers  
2.Functional Testing: Confirms consistent access and behavior during concurrent sessions

Tech Techniques :-  
1.Scenario-Based Testing Mimics a real-world case of a user logging in on multiple devices  
2.State Transition Testing Track session state between devices and browsers  
3.Exploratory Testing Check for unexpected behavior, race conditions, or session overrides

Test Method :-  
Manual Testing Use two different browsers (e.g., Chrome & Firefox) or devices (PC + mobile) simultaneously

Steps :-

1. A registered and active user account exists (e.g., test\_user).  
   2.The application is accessible from multiple browsers and/or devices (e.g., PC, laptop, mobile browser).  
   3.Two different browsers or devices are available (e.g., Chrome on PC, Firefox on mobile or another desktop).  
   4.Internet connection is active on both devices.

Pre condition:-

1. A registered and active user account exists (e.g., test\_user).  
   2.The application is accessible from multiple browsers and/or devices (e.g., PC, laptop, mobile browser).  
   3.Two different browsers or devices are available (e.g., Chrome on PC, Firefox on mobile or another desktop).  
   4.Internet connection is active on both devices.

Test data :-

Username : Admin  
Password : admin123

Severity :- High :- Allowing multiple session might be security concern

Priority :High :- Important for defining session management behaviour clearly based on apps use case and security policy.

Expected Result :-

1.Option A – If application supports multiple concurrent logins:

2.User should be able to log in and stay active on both devices/browsers simultaneously.

3.No session termination or warnings should occur.

4.Actions performed in one session should not impact the other session.

5.Option B – If application restricts concurrent logins (single active session policy):

6.Logging into Browser B should terminate the session in Browser A.

7.Browser A should show a session expired message or redirect to login.

8.Only the most recent session should remain active.

Actual result :-

1.If multiple sessions are allowed:

2.User is able to stay logged in on both Chrome and Firefox simultaneously. No session conflict or termination occurred. Actions on both sessions function independently.

3.If session is not terminated but actions cause conflicts:

4.Logged in on both browsers, but performing actions (e.g., updating profile) in one session causes the other to crash or show errors.

5.If single-session is enforced but not properly handled:

6.User logs in on second browser, and first session is not terminated, but starts malfunctioning or silently stops receiving updates.

--------------------------------------------------------------------------------------------------------------------------------------

1. Validate application behavior after clearing cookies for the domain

Bug Description:-After clearing cookies, refreshing the page redirected the user to the login page with a message: "Session expired. Please log in again."

clearing cookies invalidates the session and logs the user out.

Test Type :-

1.Functional Testing Verify system response to cookie removal  
2.Security Testing Prevent unauthorized session reuse  
3.Session Management Testing Validate session and authentication states

Test Technique :-

1.Scenario-Based Testing Simulate user clearing cookies mid-session or before login  
2.Negative Testing Induce cookie deletion to test system robustness  
3.State Transition Testing Session with cookies → No cookies → System response

Test Method :-  
Manual Testing Use browser developer tools or settings to clear cookies for the domain

Steps :-

1.Open the browser and navigate to the application login page.

2 Log in using valid credentials (Admin / admin@123).

3 Confirm login is successful and user lands on the home/dashboard page.

4 Open browser developer tools or settings and clear all cookies for the application’s domain.

5 Go back to the application tab and try to refresh the page or navigate to another protected page.

6 Observe whether the session is still active or if the application logs the user out.

Pre condition:-

1.The user must have a valid account and login credentials.

2.The application must use cookies for session management.

3.A browser that allows manual cookie clearing (e.g., Chrome or Firefox) is available.

4.Developer tools or browser settings access is enabled.

Test data :-

Username :Admin  
password : admin123

domain :- <http://orangeHRM.com>

Severity :- High:-if session is not invalidated after clearing cookies , leads to undefined behaviour

Priority :High :- This is critical for applications that rely on cookie based session for authentication

Expected Result :-

1.Clearing cookies should remove session identifiers (e.g., session\_id, auth\_token).

2.Upon refresh or navigation:

3.The application should redirect to the login page.

4.Optionally, a "Session expired" or "You have been logged out" message is displayed.

5.No protected routes or data should be accessible after cookies are cleared.

Actual result :-

1.If behavior is correct:

2.After clearing cookies, refreshing the page redirected the user to the login page with a message: "Session expired. Please log in again."

3.If session remains active (unexpected):

4.After clearing cookies, the user was still able to access the application and navigate protected content. Session was not invalidated properly.

5.If partial behavior or error occurs:

6.After clearing cookies, the application showed a blank screen or error (e.g., 401 Unauthorized) instead of gracefully redirecting to login.

--------------------------------------------------------------------------------------------------------------------------------------

1. Validate Username field not case-sensitive

Bug Description:- Login succeeds

Test type :-  
1.Functional Testing :Confirms expected login behavior  
2.Usability Testing :Ensures user-friendly input handling  
3.Negative Testing :If case sensitivity is enforced (rare), test incorrect casing rejection

Test Technique :-

1.Equivalence Partitioning :-Group all case variants as valid input  
2.Boundary Value Testing :-Use edge-case combinations (e.g., mixed casing)  
3.Scenario-Based Testing :-Simulate users typing the username in varying styles

Test Method :-  
Manual Testing Enter the username with different casing manually

Steps :-

1.Open OrangeHRM login page  
2.Enter aDmin in username  
3.Enter valid password  
4.Click Login

Pre condition:-

Valid user exists (e.g., “Admin”)

Test data :-

Username: aDmin (instead of correct Admin)  
Password: correct password.

Severity :medium

Priority :medium

Expected Result :- Username match should be case-sensitive

Actual Result :- Login succeeds

--------------------------------------------------------------------------------------------------------------------------------------

1. Validate Password field not masked / lacks "eye" toggle

Bug Description:- Password displayed in cleartext No “eye” toggle icon in Chrome, Firefox, Opera

Test type :-  
1.UI/UX Testing :Verifies visual and interactive elements  
2.Functional Testing :Confirms toggle behavior

Test Technique :-  
1.UI Element :Inspection Look at HTML input type and toggle behavior  
2.Exploratory Testing :Try enabling/disabling visibility  
3.State Transition Testing :Switch between masked and unmasked states

Test Method :-  
1.Manual Testing :Open the login form, inspect password field visually and functionally  
2.Optional: DevTools Inspection Confirm input type="password" and toggle changes it to "text" and back

Steps :-

1.Open the OrangeHRM login page  
2.Enter password  
3.Observe visibility and presence of toggle icon

Pre condition:-

Login page open in Chrome / Firefox / Opera

Test data :-

Username: valid  
Password: visible characters

Severity :high

Priority :high

Expected result :- Password masked; optional eye icon toggles visibility

Actual Result :- Password displayed in cleartext No “eye” toggle icon in Chrome, Firefox, Opera

--------------------------------------------------------------------------------------------------------------------------------------

1. Validate language selector for global users.

Bug Description:- No language dropdown or detection

Test Type:-  
1.Localization Testing :Validate text translations, labels, formats  
2.Internationalization (i18n): Ensure app supports multiple languages  
3.UI/UX Testing :Check visual layout in different languages  
4.Functional Testing :Confirm language selection updates UI correctly

Test Technique :-  
1.Equivalence Partitioning Group supported languages and test one from each category  
2.Scenario-Based Testing Mimic real-world usage (e.g., user switches to Spanish and logs in)  
3.Boundary Testing Try non-supported or invalid languages (if applicable)

Test Method :-  
Manual Testing Switch languages manually and observe changes

Steps:-1.Open OrangeHRM login Page

2.Visit the login page

Pre condition:-

Login page opened from non-English IP

Test data :- User speaks a language other than English

Severity :low ,

Priority :medium

Expected result :- Provide option for multiple languages or auto-detect

Actual Result :- No language dropdown or detection

--------------------------------------------------------------------------------------------------------------------------------------

1. Validate Lack of dark mode support

Bug Description:-Page stays bright white, blinding at night

or whether OrangeHRM supports dark mode via system settings or in-app toggle, and how the interface behaves when users expect a dark theme.

Test type :-  
1.UI/UX Testing Evaluates interface adaptability and design responsiveness  
2.Accessibility Testing Checks for visual comfort and contrast in different modes  
3.Compatibility Testing Validates responsiveness to system/browser features

1.Configuration Testing : Change OS/browser themes and check app response  
2.Visual Inspection Observe color schemes, contrast, and layout behavior  
3.CSS/DevTools :-Check Confirm presence/absence of prefers-color-scheme media query or theme toggles

Test Method :- Manual Testing Change dark mode settings in system/browser and observe OrangeHRM

Steps:-

1.Switch OS/browser to dark mode  
2.Open OrangeHRM login page

Pre condition:-

Browser/system in dark mode

Test data :- N/A

Severity :low

Priority :low

Expected Result :- Auto-adapt or offer a dark mode toggle

Actual result :- Page stays bright white, blinding at night

--------------------------------------------------------------------------------------------------------------------------------------

1. Validate that the "Forgot your password?" link on the login page is clearly visible and styled to look like an interactive element (link or button), ensuring users can easily find and use it.

Bug Description:- UI remains in light theme only, even when system/browser dark mode is active

Test Type :-  
1.UI/UX Testing :-Evaluates link visibility and style  
2.Accessibility Testing:- Confirms keyboard access, contrast, focus indicators  
3.Functional Testing :-Verifies navigation to password reset process

Test Technique :-  
1.Visual Inspection Check appearance of the link visually  
2.Exploratory Testing Interact with the link using mouse and keyboard  
3.Style Verification Check CSS for link styling and interactivity

Test Method :-  
Manual Testing Use mouse and keyboard to evaluate visibility and behavior

Test steps:-

1.Enable dark mode at the operating system level or browser preference:  
2.For Chrome: Go to system settings → Appearance → Dark.  
3.Open the OrangeHRM login page:  
<https://opensource-demo.orangehrmlive.com/>  
4.Observe the interface:  
-Check if colors adapt to dark mode.  
-Observe background, text, input fields, and buttons.  
5.Log in with test credentials (Admin, admin123).  
6.Check the dashboard and inner pages for any change in theme.  
7.Look for:  
-Any theme switch toggle inside the OrangeHRM UI.  
-CSS media query usage (prefers-color-scheme: dark).

Pre condition:-

1.Browser and operating system support dark mode (e.g., Windows/macOS in dark theme).

2.OrangeHRM is accessed on a modern browser (Chrome, Firefox, Safari, Edge).

[3.No](http://3.No) browser extensions forcing dark mode are enabled.

Test data :- OS Setting :Dark mode enabled at system /browser level.

User login :- Username : Admin , password : admin123

Severity :- Medium: security or functional issue ,but it Affects user comfort, especially for users in low - light environment ability and may frustrate users who cannot find password reset option easily .

Priority :- Low (optional/Enhancement) :Can be addressed in future UI/UX improvements , depending on product roadmap.

Expected Result :-

OrangeHRM should:

1.Detect system/browser dark mode and apply a dark theme automatically OR

2.Provide a manual toggle (e.g., 🌙 icon or "Dark Mode" switch) in the UI.

3.Text and elements should remain readable and visually accessible in dark mode

Actual Result :-

1.No dark mode support observed.

2.UI remains in light theme only, even when system/browser dark mode is active.

3.No in-app dark mode toggle provided.

4.No usage of prefers-color-scheme detected in page CSS.

--------------------------------------------------------------------------------------------------------------------------------------

1. Validate if the application implements CAPTCHA, account lockout, or any other bot protection mechanism after repeated failed login attempts.

Bug Description:- No account lockout or warning messages

1.Negative Testing: Tries invalid scenarios (wrong credentials)  
2.Functional Testing: Ensures CAPTCHA or lockout mechanisms function as expected

Test technique :-  
1.Error Guessing Repeated failed login attempts  
2.Exploratory Testing Trigger rate limits or lockout  
3.State Transition Testing Move from normal to blocked/login-restricted state

Test Method :-  
1.Manual Testing :Enter wrong credentials repeatedly and observe app behavior  
2.DevTools Inspection: Check for hidden CAPTCHA or failed login count mechanisms

Steps :-

1.Open OrangeHRM login page in browser .  
2. Enter valid username and incorrect password  
3. click Login button  
4.Observe the response "Invalid credentials" should appear  
5. repeat steps atleast 5- 10 times rapidly using different wrong password .  
6. monitor UI and behaviour for any following  
CAPTCHA appearing  
Account temporarly locked  
delayed login responses  
Error messages indicating suspicious behaviour

Pre condition:-

OranageHRM login page is accessible  
Javascript is enabled.

CAPTCHA (if implemented ) is not initially visible on login page.  
User has access to the browser or automation tool.

Test data :-

Username : admin  
Password : invalid values  
attempts : 5-10 consecutive failed attempts

Severity :- high:- comprmise account if usernames are known

Priority :- High:- This is core security gap must be addressed to comply with best practices and OWASP Top 10

Expected Result :-

1.After threshold number of failed login attempts (3-5) system should trigger at least one of the following

2.CAPTCHA challenge reCaptcha)  
Temporarly lockout (eg : too many failed attempts )

3.Login rate limiter (e.g , increasing delay )  
A warning message about suspicious login activity

Actual Result :-

1.No captcha triggered after 10+ failed attempts  
2.No account lockout or warning messages  
3.Login remains responsive , allowing repeated incorrect attempts

--------------------------------------------------------------------------------------------------------------------------------------

1. Validate whether OrangeHRM functions correctly and displays properly in Legacy Browsers

Bug Description:- The password field appears filled ( with dots ) but the JS code return an empty string or " " until user interacts manually

1.Compatibility Testing :Checks behavior across browser versions  
2.UI/UX Testing :Validates layout and visual appearance  
3.Functional Testing :Ensures core features work (e.g., login, navigation)

Test technique :-  
1.Cross-browser testing: Manually or using tools like BrowserStack  
2.Visual Inspection :Observe layout, responsiveness, fonts, icons  
3.Functional walkthrough: Try critical user journeys (login, menu clicks, etc.)

Test method :- Manual Testing Use legacy browsers directly (locally or via virtual machine)

Steps:-

1.launch legacy browser (google chrome ) with autofill /password manager is enabled.  
2. Navigate to OrangeHRM url  
3. click on username field - browser should auto-suggest credentials.  
3. select the saved credentials from autofill popup.  
4. Once autofilled , do not submit the form.  
5. Open developer console (F12) run this  
document.querySelector('input[type="password"]').value  
6. Observe whether password is visible through the script .

Pre condition:

1.Legacy browser is installed  
2.internet connection available  
3.javascript enabled  
4.OrangeHRM login URL is accessible  
5.Priority: Low  
6.Severity: Minor (Compatibility)

Test data :-

Username :Admin  
Password :admin123

Legacy Browsers to Consider:-  
Browser Version Notes  
1.Internet Explorer 11 or below End-of-life, but still used in some enterprises  
2.Chrome v49 or below Pre-auto-update versions (Windows XP)  
3.Firefox v52 or below Older UI, slower JS engine  
4.Safari v9 or below Older macOS compatibility  
5.Edge (Legacy) Pre-Chromium versions Not supported by modern apps

Severity :- High :-if javascript can access autofilled password values without user action.

Priority :- High :- must be addressed immediately for any production environment

Expected Result :-

1.Password filled may be auto filled, but its value should not be accessible via Javascript before form submission.

2.Browser should protect autofilled password values from being read by javascript unless explicitly interacted with by the user .

Actual Result :-

1.Vulnerable behaviour(Fail) :- Running the script above returns the plain text password (admin123) meaning auto filled password is exposed to the DOM and accessible via JS

2.Secure behaviour(not exposed pass) :- The password field appears filled ( with dots ) but the JS code return an empty string or " " until user interacts manually

--------------------------------------------------------------------------------------------------------------------------------------

1. Validate Input contains HTML like <script> or <b>.

Bug Description:- Input was reflected back without escaping

To validate login input field properly handle or sanitize HTML content to prevent cross site scripting or markup injection attack.

Test type :-  
1.input Validation Testing :-Validate server-side and client-side sanitization  
2.Functional Testing :-Check whether inputs are handled properly in forms

Test Technique :-  
1.Error Guessing :Use known malicious or malformed inputs  
2.Injection Testing :Try inserting HTML/JS payloads  
3.Boundary Testing :Enter edge-case inputs like <script></script>, long HTML tags, etc.

Test Method :-  
1.Manual Testing Insert HTML tags into form fields and observe behavior  
2.Optional: Burp Suite/ZAP For security-level input fuzzing and script injection  
3.Optional: Browser DevTools Inspect how the input is rendered or stored in the DOM

Test Steps:-

1.Open OrangeHRM login page  
2. Username field enter <script> alert ('XSS')</script>  
3. in the password field , enter any dummy value  
4. Click login button press Enter  
5. Observe :  
is an alert box triggered  
is any content rendered in HTML format ?  
6. Repeat the test using this alternative value in username field :  
<b> admin </b>  
7. Check if rendered UI displays "Admin" in bold or any visual difference

Pre condition:-

1)The user is on the login page, and no user is currently logged in.  
2)Username and password input fields are visible and enabled.  
3)The application does not have front-end or back-end input sanitization mechanisms in place (or their behavior is being tested).  
4)The browser must allow JavaScript execution (to observe any <script> tag effects).  
5)Test environment should allow for potential security testing (e.g., not a production environment).

Test data :-

Username:- <script>alert('XSS')</script>admin  
Password:- </b>admin123

Severity :high

Priority :high

Expected Result :-

1.Application should not render or execute any HTML or Javascript content.

2.Input values should be treated as plain text

3.No alert boxes should appear

4.No HTML formatting should take effect.

Actual Result :-

1.Input was reflected back without escaping

2.<b> tag rendered the text in bold - indicating HTML injection

3.No script executed , but potential for reflected XSS if unescaped.

----------------------------------------------------------------------------------------------------------------

1. Validate if javascript is able to capture or modify clipboard contents during paste event in the login form (username or password field )

Bug Description:- Alert box confirmed manipulation.

Test type :-  
1.Privacy Testing :Ensures user input is not silently altered or logged  
2.Functional Testing: Verifies normal paste behavior is preserved

Test techniques :-  
1.Event Monitoring :Inspect JavaScript listeners for paste or input events  
2.DevTools Inspection: Watch clipboard behavior during form input

Test Method :-  
1.Manual Testing :Paste content and observe any behavior visually  
2.Browser DevTools :Inspect for active JavaScript listeners or clipboard manipulation

Test steps :-

1.Open the browser’s Developer Console (F12 or right-click → Inspect).  
2.Navigate to the login page (e.g., OrangeHRM).  
[3.In](http://3.In) the Console, inject the following JavaScript to simulate a clipboard hijack:  
document.querySelector('input[type="password"]').addEventListener('paste', function(e) {  
e.preventDefault();  
const hijacked = 'hackedPassword';  
e.target.value = hijacked;  
alert("Clipboard data modified!");  
});  
4.Copy a known password to your clipboard (e.g., admin123).  
5.Paste the value into the password field using Ctrl + V or right-click → Paste.  
6.Observe the behavior:  
Does the pasted value get modified or replaced?  
Is any alert or message triggered?

Pre condition:-1.Access to login page

2.Developer console access (browser dev tool)

3.browser support javascript

4.Clipboard contains known value to test modification.

Test data :-

Clipboard :admin123 Username : pasted from clipboard  
password : pasted from clipboard

Severity:-

Critical :modify password  
logging in with false data.  
Violate browser security expectations

Priority :- High :- Should be addressed immediately  
sanitizing input  
following secure coding practices

Expected Result :- Clipboard data should not be modified on paste.

Actual result :-

1.JavaScript was able to intercept the paste event and modify the input field with hackedPassword.

2.Alert box confirmed manipulation.

3.Clipboard hijack confirmed.

--------------------------------------------------------------------------------------------------------------------------------------

1. Validate whether the “Remember Me” checkbox is accessible using only keyboard navigation (Tab/Enter).

Bug Description:- Checkbox was not reachable using Tab key

Test type :-  
1.UI/UX Testing Confirms visibility and interaction cues  
2.Functional Testing Validates correct state behavior of the checkbox

Test Technique :-  
1.Keyboard Navigation Testing Use only keyboard (Tab, Enter, Space)  
2.Visual Inspection Watch for focus indicator and state changes  
3.State Testing Confirm checked/unchecked values toggle properly

Steps:- 1.Open OrangeHRM login page in browser.  
2.Press Tab key repeatedly to navigate through the input field.  
3.Observe the Focus order :  
focus should move the username -> password ->Remember me checkbox ->Login button 4.when focus reaches "Remember Me " checkbox

5.press the space key to select /deselect it.

6.complete login by tabbing to the login button and pressing enter

Pre condition:-

1.User has access to OrangeHRM login page  
2.Browser is functional ,and javascript is enabled  
3.Only keyboard is used during testing.

Test data :-

Username : Admin  
password : admin123

Severity :- High :- Accessibility issue, keyboard only users cannot select "Remember me " option .

Priority :- Medium to high :- Should be fixed soon for compilance with accessibility standards and usability best practices.

Expected Result :-

1. "Remember me " checkbox should recieve focus via Tab Key

2."Check box should be selectable using space key .

3.all login elements (username, password, checkbox, login button)must be keyboard accessible

Actual result :-

1.Checkbox was not reachable using Tab key .  
2.Focus skipped directly from password to login button.

3.Checbox could not be selected without mouse interaction.

--------------------------------------------------------------------------------------------------------------------------------------

1. Validate whether the “Remember Me” checkbox is accessible using only keyboard navigation (Tab/Enter).

Bug Description:- Focus skipped directly from password to login button.

Test type :-  
1.UI/UX Testing Confirms visibility and interaction cues  
2.Functional Testing Validates correct state behavior of the checkbox

Test Technique :-  
1.Keyboard Navigation Testing Use only keyboard (Tab, Enter, Space)  
2.Visual Inspection Watch for focus indicator and state changes  
3.State Testing Confirm checked/unchecked values toggle properly

Steps:- 1.Open OrangeHRM login page in browser.  
2.Press Tab key repeatedly to navigate through the input field.  
3.Observe the Focus order :  
focus should move the username -> password ->Remember me checkbox ->Login button when focus reaches "Remember Me " checkbox press the space key to select /deselect it.complete login by tabbing to the login button and pressing enter

Pre condition:-

1.User has access to OrangeHRM login page  
2.Browser is functional ,and javascript is enabled  
3.Only keyboard is used during testing.

Test data :-

Username : Admin  
password : admin123

Severity :- High :- Accessibility issue, keyboard only users cannot select "Remember me " option .

Priority :- Medium to high :- Should be fixed soon for compilance with accessibility standards and usability best practices.

Expected Result :-

1. "Remember me " checkbox should recieve focus via Tab Key

2."Check box should be selectable using space key .

3.all login elements (username, password, checkbox, login button)must be keyboard accessible

Actual result :-

1.Checkbox was not reachable using Tab key .  
2.Focus skipped directly from password to login button.

3.Checkbox could not be selected without mouse interaction.

--------------------------------------------------------------------------------------------------------------------------------------

1. Validate successful login using correct credentials and back-end SQL query validation.

Bug description:- user not found, or database not updated.

Test type :-  
1.Functional Testing Validates the login flow  
2.Back-End (Database) Testing Checks DB state and queries  
3.Integration Testing Confirms app <-> DB interaction correctness  
4.Positive Testing Uses correct, expected input

Test techniques :-

1.Input Validation Use valid username/password  
2.Database Query Testing Verify query logic and values in the DB  
3.Session Validation Confirm session/cookies created post-login

Test Method (Manual + DB Validation):-  
Pre-requisites:  
1)Access to the OrangeHRM database (e.g., via MySQL Workbench, pgAdmin, or SQL CLI)  
2)Known valid credentials  
3)Knowledge of the user table schema (commonly hs\_hr\_users, ohrm\_user, or similar)

Steps:-

1.Open the OrangeHRM login page.  
2.Enter Username: Admin  
3.Enter Password: admin123  
4.Click the Login button.  
5.Check if user is redirected to the dashboard.  
6.On the database side, execute:  
SELECT \* FROM users WHERE username = 'Admin';  
7.Validate that password entered matches the hashed password stored (compare in app logic or manually for test).  
8.Check if last\_login field is updated in DB:  
SELECT last\_login FROM users WHERE username = 'Admin';  
9.Check if a record is added in the login log table (if applicable):  
SELECT \* FROM login\_audit\_log  
WHERE username = 'Admin'  
ORDER BY login\_time DESC LIMIT 1;

Pre condition:-

1.User Admin exists in the database with a valid hashed password.  
2.User is active and not locked or disabled.  
3.Database and application server are running.  
4.Access to OrangeHRM login page:  
<https://opensource-demo.orangehrmlive.com/>

Test data :- Username : Admin  
Password : admin123

Severity :- Medium – Basic functionality; not logging in affects user access but not a system crash.

Priority :- High – Login is a core feature, must work reliably for all users.

Expected result :-

1.User is logged in successfully and lands on the dashboard.

2.Login credentials match database values.

3.last\_login timestamp is updated.

4.A login record is added to login\_audit\_log.

5.No errors or exceptions are raised.

Actual result :-

Pass – All actions performed as expected and backend queries confirm data updates.

(You can change to Fail if login fails, user not found, or database not updated.)

--------------------------------------------------------------------------------------------------------------------------------------

1. Validate that each login attempt (successful or failed) is logged in the login\_audit\_log table.

Bug Description:- timestamp is not updated or remains unchanged after login.

Test Type :-

1.Functional Testing Checks application logic for updating login time  
2.Back-End Testing Validates changes in the database

Test technique :-  
1.State Transition Testing :From “not logged in” to “logged in” triggers DB update  
2.Database Inspection :-Manually query user info before and after login  
3.Timestamp Validation :-Compare timestamps to system time

Pre requisites :-  
1.Access to the database (MySQL/PostgreSQL etc.)  
2.Knowledge of the user table (ohrm\_user, users, or similar)  
3.Identify the correct field: last\_login, login\_time, or in a related audit/log table

Steps :-

1.Connect to the database and note the current value of last\_login for the user:  
SELECT last\_login FROM users WHERE username = 'Admin';  
2.Note the value (e.g., 2025-06-27 08:42:00).  
3.Wait 1–2 minutes (to observe timestamp difference clearly).  
4.Open the OrangeHRM login page:  
<https://opensource-demo.orangehrmlive.com/>   
5.Log in with:  
Username: Admin  
Password: admin123  
6.After successful login, run the SQL query again:  
SELECT last\_login FROM users WHERE username = 'Admin';  
7.Compare the new timestamp to the previous one.

Pre condition:-

1.User Admin exists and has valid login credentials.  
2.Access to the OrangeHRM database (e.g., MySQL or PostgreSQL).  
3.Application server is running.  
4.You have read access to the users or equivalent table containing last\_login.

Test data :-

Username : Admin  
Password : admin123

1.Query Timestamp Before Login  
SELECT user\_name, last\_login FROM ohrm\_user WHERE user\_name = 'testuser';

2.Perform login  
1)Open OrangeHRM  
2)Login as 'testuser' with valid credentials  
3)Wait for redirection to the dashboard/home

3.Query Timestamp After Login:-  
SELECT user\_name, last\_login FROM ohrm\_user WHERE user\_name = 'testuser';  
Capture updated timestamp (T2)

4.Compare Results  
Check Expectation  
T2 > T1, Timestamp is updated  
T2 ≈ current system time Reflects actual login moment  
No change DB update failed or misconfigured

Severity :-Medium – Impacts auditability and user activity tracking, but not core functionality.

Priority :- Medium – Should be addressed for compliance, reporting, and security monitoring.

Expected Result :-

1.The last\_login field for user Admin should be updated to the current datetime after successful login.

2.The new value should be greater than the old timestamp.

Actual Result :-

Pass – If last\_login is updated correctly with the current login time.  
Fail – If the timestamp is not updated or remains unchanged after login.

--------------------------------------------------------------------------------------------------------------------------------------

1. Validate Count Login Attempts in the Last 24 Hours Verify the system correctly logs and can count all login attempts for a user within the last 24 hours using SQL.

Bug description:- Count does not match expected attempts

Test type :-  
1.Database Testing :-Query-based validation of log records  
2.Audit Testing :-Verifies proper logging of user actions  
3.Functional Testing :-Confirms log consistency with actions

Test technique :-

1.Database Query Testing Query login log tables  
2.Date/Time Validation Use SQL time filters (NOW() - INTERVAL 1 DAY)  
3.Positive Testing Count logs for a real user

Assumption and setup :-

1.Login log table: login\_audit\_log (or similar, e.g., ohrm\_login\_activity)  
2.Relevant fields:  
username  
timestamp (or login\_time)  
status (SUCCESS, FAILURE)

Steps:-

1.Open SQL client (phpMyAdmin, MySQL CLI, pgAdmin, etc.).  
2.Log in to the OrangeHRM application as Admin multiple times (some successful, some with wrong passwords if possible).  
3.Run the following SQL query to count login attempts for the last 24 hours:  
SELECT COUNT() AS login\_attempts  
FROM login\_audit\_log  
WHERE username = 'Admin'  
AND login\_time >= NOW() - INTERVAL 1 DAY;  
4.SQL Query to Retrieve All Login Entries in Last 24 Hours (Optional)  
SELECT username, timestamp, status  
FROM login\_audit\_log  
WHERE username = 'testuser'  
AND timestamp >= NOW() - INTERVAL 1 DAY  
ORDER BY timestamp DESC;  
5.Optionally, verify total success vs. failed attempts:  
SELECT success, COUNT()  
FROM login\_audit\_log  
WHERE username = 'Admin'  
AND login\_time >= NOW() - INTERVAL 1 DAY  
GROUP BY success;

Pre condition:-

1.User Admin exists.  
2.Login audit logging is enabled (i.e., attempts are stored in a table like login\_audit\_log).  
3.Tester has read access to the OrangeHRM database.

Test data :- Field Value  
Username Admin  
Timeframe Last 24 hours

Severity :- Medium – Affects audit trails and brute-force detection, but not core login functionality.

Priority :- Medium to High – Important for security audits and suspicious login monitoring.

Expected Result :-

1.The COUNT(\*) result should reflect the number of times the user Admin attempted to log in during the last 24 hours.

2.If you attempted 5 logins (3 successful, 2 failed), you should see login\_attempts = 5.

3.Optional: success count = 3, fail count = 2.

Actual Result :-

Pass – If the count matches the number of login attempts made in the last 24 hours.  
Fail – If:  
No records are returned  
Count does not match expected attempts  
Timestamps are wrong or not logged

1. Validate by Capture the current value of the last\_login field for a specific user before and after login is attempted.

Bug description:-last\_login column is missing or shows invalid data.

Test type and Technique :-  
1.Database Testing SQL-based data state validation  
2.State Transition Testing Validating a change after event  
3.Functional Testing Application login behavior

Failed Login attempt :-  
1.Try logging in with an invalid password.  
2.Re-run the after-login query.  
3. last\_login should not change.

Steps:-

Test Steps:  
1.Open your database client (e.g., MySQL CLI, phpMyAdmin).  
2.Run the following SQL query:  
SELECT last\_login  
FROM users  
WHERE username = 'Admin';  
4.Note and save the value of the last\_login field before performing login.  
(In a full test case, you would then log in and check if this value changes.)

Step 1: Capture last\_login Before Login  
SELECT user\_name, last\_login  
FROM ohrm\_user  
WHERE user\_name = 'testuser';  
Record the timestamp as T\_before.

Step 2: Perform Login (Web or API)  
1.Login via the application using:  
2.Username: testuser  
3.Password: ValidPassword123  
4.Wait for dashboard to load (ensure it’s a successful login)

Step 3: Capture last\_login After Login  
SELECT user\_name, last\_login  
FROM ohrm\_user  
WHERE user\_name = 'testuser';  
Record the timestamp as T\_after.

Step 4: Compare Results  
Condition Expected Outcome  
T\_after > T\_before last\_login updated  
T\_after ≈ current time Reflects actual login time  
T\_after = T\_before Login didn’t update the DB

Add Timestamp Delta Check:-  
SELECT TIMESTAMPDIFF(SECOND, last\_login, NOW()) AS seconds\_since\_last\_login  
FROM ohrm\_user  
WHERE user\_name = 'testuser';  
Expect result to be a few seconds or less if tested immediately.

Pre condition:-

1.The user Admin exists in the users table.  
2.You have read access to the OrangeHRM database.  
3.OrangeHRM login tracking (i.e., last\_login column) is implemented.

Assumptions:-  
1.The user table is ohrm\_user (common in OrangeHRM)  
2.The field is named last\_login (may also be login\_time in some schemas)  
You have:  
3.Valid username: 'testuser'  
4.DB access (e.g., MySQL Workbench or CLI)  
5.Access to login form or API

Test data :- Field Value  
Username Admin

Severity :- Medium – Affects audit/log tracking, not login itself.

Priority :- Medium – Important for analytics and security audits.

Expected Result :- The query should return one timestamp value or NULL (if the user has never logged in).

Format should be a valid datetime (e.g., 2025-06-27 08:42:17).

1.Before Login:

username \_lastlogin

testuser | 2025-07-20 14:23:11

1.After Login:  
testuser 2025-07-21 10:17:55 This confirms last\_login was updated after login.

Actual result:-

Pass – If:  
A valid datetime value or NULL is returned  
Fail – If:  
No record is returned  
last\_login column is missing or shows invalid data

Employee List :-

1. Validate that an employee who is scheduled to be terminated in the future still appears when Include = "Current Employees Only" is selected in the search filter

Bug Description:- The search was executed with Include = "Current Employees Only".

The employee with a future termination date did not appear in the search results.

Test type :-  
1.Boundary Value Analysis (BVA): Since the employee is on the edge of becoming a past employee (termination is scheduled but not yet effective), this is a boundary condition between "Current" and "Past" status.

2.Equivalence Partitioning :Classifies employees into logical groups (Current, Scheduled for Termination, Past). The test validates that employees in the "Scheduled for Termination" partition are treated as Current.

3.Negative Testing Ensures the system does not wrongly exclude an employee who is still technically current.  
Decision Table Testing (optional) Helpful if you have multiple filters interacting, and want to validate combinations of Include + Termination Date + Employment Status.

4.Decision Table Testing (optional) :Helpful if you have multiple filters interacting, and want to validate combinations of Include + Termination Date + Employment Status.

Test method :- Method Description  
Manual Testing Use the UI to apply the filter and observe the behavior directly. Validate whether the employee is shown based on the filter logic.

Steps :-

1.Go to <https://opensource-demo.orangehrmlive.com>   
2.Log in using Admin credentials:  
Username: Admin  
Password: admin123  
3.Navigate to PIM > Employee List.  
[4.In](http://4.In) the Employee Name field, enter Linda Anderson and select from the suggestion.  
5.From the Include dropdown, select Current Employees Only.  
6.Leave other filters blank or as default.  
7.Click the Search button.  
8.Observe whether the employee record is returned.

Pre condition:-

\*User is logged into OrangeHRM Demo as Admin.  
\*Employee Linda Anderson exists and is assigned a future termination date (via Admin > Job > Terminate Employment).  
\*Employee is still considered "current" as of today's date.

Test data :-

Employee Name: Linda Anderson  
Employment Status: Full-Time Permanent  
Termination Date: A future date (e.g., 2 weeks from today)  
Include: Current Employees Only

Severity :medium

Priority :High

Expected Result :- \*Linda Anderson should be shown in the search results.

Employee should be still considered active , because termination date is in future.  
\*Since her termination date is in the future, she is still considered a current employee.

Negative/Edge Case Scenarios:-  
Scenario :Expected Outcome  
1.Termination date is today :Employee may or may not show (depends on time logic; confirm business rule)  
2.Termination date is in the past :Employee should not be shown  
3.Termination date is missing/null Treated as current employee

Actual Result :- Pass :- \* The OrangeHRM demo system does not support scheduling future terminations via the UI in the current public version.  
However, in a custom or enterprise instance, if a future termination is configured:

The system should still list the employee under Current Employees Only until the termination date is reached.

Fail :- The search was executed with Include = "Current Employees Only".

The employee with a future termination date did not appear in the search results.

--------------------------------------------------------------------------------------------------------------------------------------

1. Validate that using partial filters like Job Title and Sub Unit correctly returns all matching employee records.

Bug Description:- No Records Returned.

Test type :-

1.Functional Testing :Validates core filter functionality across multiple fields.  
2.Integration Testing: Ensures Job Title and Sub Unit filters work together correctly.  
3.Regression Testing :Verifies that previous functionality isn’t broken after updates.  
4.Equivalence Partitioning :Ensures all employees in a category (job title/sub unit) are returned.  
5.Boundary Testing (light) Tests combinations where results are 0, 1, or >10 records.

Test Method :-  
Manual Testing :Use the UI to apply partial filters (Job Title + Sub Unit) and validate the results.

Test Data:-  
Employee Name Job Title Sub Unit  
1.Alice Johnson QA Analyst IT Department  
2.David Kim QA Analyst IT Department  
3.Emma Stone QA Analyst Finance  
4.John Lee Developer IT Department

Scenarios to Test :-  
1.Job Title = QA Analyst Alice, David, Emma  
2.Sub Unit = IT Department Alice, David, John  
3.Job Title = QA Analyst + Sub Unit = IT Department Alice, David  
4.Job Title = Developer + Sub Unit = Finance No results

Steps :-

1.Go to <https://opensource-demo.orangehrmlive.com>   
2.Log in with Admin credentials:  
Username: Admin  
Password: admin123  
3.Click PIM from the left-hand navigation menu.  
4.On the Employee List page:  
5.From the Job Title dropdown, select HR Manager.  
6.From the Sub Unit dropdown, select Human Resources.  
7.Leave all other fields blank.  
8.Click on the Search button.  
9.Observe the results in the employee list.

Pre condition:-

1.User is logged into OrangeHRM Demo as Admin.  
2.At least one employee (e.g., Linda Anderson) is assigned the Job Title: HR Manager and belongs to the Human Resources Sub Unit.

3.Optionally, other employees also share this combination (to verify multiple results).

Test data :-

Job Title: HR Manager

Sub Unit: Human Resources

Severity :medium

Priority :medium

Expected Result :- The system should return all employees who have the Job Title: HR Manager and are part of the Human Resources Sub Unit.  
For example: Linda Anderson and any other employees matching the filters.

1.Select mismatched Job Title + Sub Unit :No records shown  
2.Enter invalid characters in filters via dev tools Input sanitized, system doesn’t break.  
3.Filters applied, then quickly removed UI refreshes and displays correct default results

Actual Result :-

Pass :- The system displayed Linda Anderson and any other matching employees with Job Title: HR Manager and Sub Unit: Human Resources.  
Filtering worked correctly with just two criteria.

Fail :- The query or UI search executed successfully.  
One of the following issues occurred:  
Case 1: No Records Returned  
Input:  
Job Title: HR Manager  
Sub Unit: Human Resources  
No records were returned, despite knowing that employees (e.g., Linda Anderson) match both filters.  
Issue:  
Filtering logic failed to return existing data, possibly due to:  
Incorrect filter handling (e.g., wrong SQL joins).  
Data mismatch (e.g., "HR Manager" vs "Hr Manager").  
Case sensitivity or trailing spaces in stored data.

Case 2: Incomplete Results Returned  
Some matching employees were returned,  
But not all expected employees were listed.  
Example:  
Only 1 HR Manager shown, when 3 are known to exist in Human Resources.  
Possible Cause:  
Filters are applied using AND/OR improperly.  
One filter (Job Title or Sub Unit) is failing silently due to nulls or bad mapping.

Case 3: Incorrect Results Returned  
Results include employees who:  
Don’t match the Job Title, or  
Belong to the wrong Sub Unit.  
Example:  
emp\_number name job\_title sub\_unit  
0035 John Miller Accountant Human Resources  
0042 Jane Smith HR Manager IT Department

----------------------------------------------------------------------------------------------------------------

1. Validate that querying by Employee ID returns the correct employee record.

Bug Description:- Multiple Records Returned

1.Test Type: Functional Testing  
2.Test Method  
Can be performed through:  
Manual Testing (via UI or DB client)  
3.Test Technique  
Black Box Testing (focuses on inputs and outputs without internal code knowledge)  
May also include:  
1.Equivalence Partitioning (valid ID, invalid ID, empty ID)  
2.Boundary Value Analysis (e.g., smallest/largest ID formats)

Steps :-

1.Launch your SQL client or database management tool (e.g., MySQL Workbench, SQL Server Management Studio, pgAdmin).  
2.Connect to the database where the employee information is stored.  
3.Open a new SQL query window or editor.  
SELECT \* FROM employees WHERE emp\_number = '0012';  
Execute the query.  
4.Review the returned result set to confirm that exactly one record with Employee ID '0012' is displayed.  
5.Verify that all relevant employee details (such as name, job title, status, supervisor, etc.) are present and accurate.

Pre condition:-

Employee with ID '0012' exists in the database.

Test data :-

Employee ID = '0012'

Severity :high

Priority :high

Expected Result :- One employee record with Employee ID '0012' is returned and all fields (name, job title, status, supervisor, etc.) are correct and not null.

Actual Result :- Pass :- \*The query returned exactly one record for Employee ID '0012'.  
\*Employee details displayed correctly:  
Name: Linda Anderson  
Job Title: HR Manager  
Employment Status: Full-Time Permanent  
Supervisor: John Smith  
Sub Unit: Human Resources  
\*No errors occurred during query execution.  
\*Data matches expected employee information accurately.

Fail :- \* Case 1: No Record Returned  
Query:  
SELECT \* FROM employees WHERE emp\_number = '0012';  
No record was returned, even though Employee ID '0012' is expected to exist.  
This may indicate:  
Data was deleted or never inserted.  
Typo in emp\_number.  
Mismatch in data format (e.g., leading zeros).

Case 2: Incorrect Data Returned  
One record was returned, but:  
Employee name or other fields do not match expected data.  
Job title, department, or status is incorrect or outdated.  
Example Output:  
emp\_number first\_name last\_name job\_title  
0012 Linda Anderson Intern ← should be HR Manager

Case 3: Multiple Records Returned  
More than one record with the same emp\_number is returned.  
This indicates a violation of uniqueness constraint, which is critical.

----------------------------------------------------------------------------------------------------------------

1. Validate the count of employees grouped by employment status.

Bug Description:- The query executed successfully.

The returned result contains incorrect or inconsistent employee counts per status, or some expected statuses are missing.

1.Test Type: Functional Testing  
2.Test Method : Manual Testing using a reporting interface or SQL query  
3.Test Technique :   
1)Black Box Testing — observe output for given inputs without internal logic  
2)Equivalence Partitioning — test across status types (e.g., Active, Terminated, On Leave)  
3)Boundary Value Analysis — check edge cases (e.g., only one employee in a status)

Steps :-

1.Open your SQL client (e.g., MySQL Workbench, pgAdmin, DBeaver, etc.).  
2.Connect to the OrangeHRM or employee-related database.  
3.Open a new SQL query window.  
4.Execute the following SQL query:  
SELECT employment\_status, COUNT(\*) AS total  
FROM employees  
GROUP BY employment\_status;  
5.Review the output result set.  
6.Cross-check the counts manually or with a known reference dataset (if available).

Pre condition:-

\*The employee table contains multiple employee records.  
\*These records include at least two or more different employment statuses (e.g., Full-Time Permanent, Part-Time, Probation).

Test data :-

None required — this is an aggregate query using existing data in the database.

Severity :low

Priority :high

Expected Result :-1.The result should return a list of each unique employment status along with the accurate number of employees associated with each.  
2.Example Output:  
employment\_status total  
Full-Time Permanent 12  
Part-Time 5  
Probation 3  
[3.No](http://3.No) null statuses or invalid groups should be included.  
4.Query executes successfully without any errors.

Actual result :- Pass:-

The query executed successfully.

Returned the following result:  
employment\_status total  
Full-Time Permanent 12  
Part-Time 5  
Probation 3  
\*Data accurately reflects the number of employees per status as expected.

Fail :- The query executed successfully.

The returned result contains incorrect or inconsistent employee counts per status, or some expected statuses are missing.

----------------------------------------------------------------------------------------------------------------

1. Validate that all employees who have a supervisor assigned are linked to a valid existing employee record in the system (no orphan supervisor references).

Bug Description:- Query executed successfully.

One or more records were returned.

The returned employee(s) have a supervisor\_id that does not match any existing emp\_number in the employees table.

1.Test Type :- Data Integrity Testing  
(Also applies to Functional Testing if checking via API or UI)

2.Test Method :- Test Method  
Typically done via:  
1)SQL Query (for database validation)  
2)Automated Backend Tests  
3)ETL or Data Validation Scripts

3.Test Technique :- Join-Based Validation  
SELECT e.employee\_id, e.supervisor\_id  
FROM employees e  
LEFT JOIN employees s ON e.supervisor\_id = s.employee\_id  
WHERE e.supervisor\_id IS NOT NULL AND s.employee\_id IS NULL;

Steps :-

1.Open your SQL client (e.g., MySQL Workbench, DBeaver, pgAdmin).  
2.Connect to the database where the employees table is located.  
3.Open a new SQL query editor.  
SELECT e.emp\_number, e.first\_name, e.last\_name, e.supervisor\_id  
FROM employees e  
LEFT JOIN employees s ON e.supervisor\_id = s.emp\_number  
WHERE e.supervisor\_id IS NOT NULL AND s.emp\_number IS NULL;  
4. Observe the query result.  
If any rows are returned, those employees have invalid supervisor references.

Pre condition:-

\*The employees table must contain records where some employees are assigned supervisors via supervisor\_id.  
\*The referenced supervisor IDs should also exist in the emp\_number field of the same employees table.

Test data :-

\*No specific input data is needed.  
\*The query uses existing relationships in the employees table.

Severity :high

Priority :high

Expected Result :-

No records should be returned.  
This indicates that every supervisor ID assigned to an employee exists in the employees table.

Actual Result :-

Pass :- Query executed successfully.

No records were returned.  
\*This confirms that all supervisor relationships are valid and all supervisor\_id values exist in the emp\_number column.

Fail :- Query executed successfully.

One or more records were returned.

The returned employee(s) have a supervisor\_id that does not match any existing emp\_number in the employees table.

This indicates invalid or orphaned supervisor references, which could lead to reporting structure errors in the system.

1. Validate Invalid Employee ID Lookup(negative scenario)

Bug Description:- Records returned unexpectedly

Test Type:-  
1.Negative Testing  
2.Functional Testing

Test Method :-  
1.Manual Testing (through UI, API, or DB client)

Test Design Techniques :-  
1.Black Box Testing  
2.Boundary Value Analysis  
3.Error Handling Validation

Steps :-

1.Connect to the OrangeHRM database.  
2.Execute:  
SELECT \* FROM hs\_hr\_employee WHERE emp\_number = '9999';  
Check the result set.

Pre condition:-

The hs\_hr\_employee table must exist.

Test Data :-

emp\_number = '9999' (non-existent)

Severity :low

Priority :high

Expected Result :- No records returned.

Actual result :- Pass (No records returned) / Fail (Records returned unexpectedly)

-------------------------------------------------------------------------------------------------------------

1. Validate Employees with Same National ID (SSN Duplication Check)

Bug Description:- Query returned one or more records with the same ssn\_num.  
1.Test Type:Data Integrity Testing  
2. Test Method:Manual Query (ad hoc validation)  
3.Test Technique :-  
White Box Testing  
Duplicate Key Detection

SELECT ssn, COUNT() AS count  
FROM employees  
WHERE ssn IS NOT NULL  
GROUP BY ssn  
HAVING COUNT() > 1;  
Pass: Query returns 0 rows

Fail: Any results mean duplicate SSNs are present

Steps :-

Execute:  
SELECT ssn\_num, COUNT()  
FROM hs\_hr\_employee  
WHERE ssn\_num IS NOT NULL  
GROUP BY ssn\_num  
HAVING COUNT() > 1;  
Check output for duplicates.

Pre condition:-ssn\_num column must exist.

Test data :- none.

Severity :high

Priority :high

Expected Result :-

[1.No](http://1.No) records should be returned.  
2.Each ssn\_num should be unique across all employee records.

Actual Result:-

Actual Result – Pass (If No Duplicates):

Query executed successfully.  
-No records were returned.

All employee national IDs (SSNs) are unique.

Actual Result – Fail (If Duplicates Found):

Query returned one or more records with the same ssn\_num.

Indicates a data integrity issue: multiple employees are sharing the same national ID.

1. Validate Overlapping Supervisor Relationship

Bug Description:- data error — employee(s) are listed as their own supervisor." returned more than 1 record.

Test Type: Data Integrity Testing  
Test Method :- SQL or recursive logic checks in the database Techniques :- Self-Supervision Check

SELECT employee\_id  
FROM employees  
WHERE employee\_id = supervisor\_id;  
Invalid if any rows returned

🔹 Simple Circular Reference (A → B → A)

SELECT a.employee\_id AS emp1, b.employee\_id AS emp2  
FROM employees a  
JOIN employees b ON a.supervisor\_id = b.employee\_id  
WHERE b.supervisor\_id = a.employee\_id;  
Invalid if any rows returned

🔹 Recursive Loop Check (A → B → C → A)  
If your database supports Common Table Expressions (CTEs):

WITH RECURSIVE hierarchy AS (  
SELECT employee\_id, supervisor\_id, employee\_id AS root\_id  
FROM employees  
WHERE supervisor\_id IS NOT NULL

UNION ALL

SELECT e.employee\_id, e.supervisor\_id, h.root\_id  
FROM employees e  
JOIN hierarchy h ON e.supervisor\_id = h.employee\_id  
)  
SELECT \*  
FROM hierarchy  
WHERE employee\_id = root\_id;  
Any rows indicate a circular hierarchy.

Pre condition:- supervisor\_id column should exist.

Test data :- None

Severity : high

Prirority : high

Test steps :- 1.Execute:

2.Connect to the database.

Run the following SQL query:

SELECT emp\_number, supervisor\_id

FROM hs\_hr\_employee

WHERE emp\_number = supervisor\_id;

Observe the result set.

Select p\_number, supervisor\_id

FROM hs\_hr\_employee

WHERE emp\_number = supervisor\_id;

Verify no self-referential links.

Expected Result :- No records should be returned.

This indicates no employee supervises themselves

Actual result :- Actual Result – Pass (If No Records Returned)

"Query executed successfully. No records returned. This confirms there are no self-referential supervisor assignments."

Actual Result – Fail (If Records Found)

"Query returned 1 or more records where emp\_number = supervisor\_id. This indicates a data error — employee(s) are listed as their own supervisor."

**Add Employee :-**

35.Test case : Validate Add Employee Without First Name

Bug Description :- Employee was added with incomplete data.

Note :- First Name is mandatory — system should throw an error if not entered

Pre condition :- 1.User is logged in with Admin privileges.

2.User is on the PIM > Add Employee page in OrangeHRM.

Test data :- First Name: (Leave Blank)

Last Name: Doe

Severity : Medium

Priority : High

Expected Result :- Expected Result:

\*System displays a validation error:

"Required" or "First Name is required".

\*The form is not submitted, and no employee is created.

Actual result :-

Actual Result – Pass:

System did not allow submission.

Error message "Required" appeared below the First Name field.

User remained on the same page.

Result: Pass

Actual Result – Fail (if it happened):

The form submitted without First Name.

Employee was added with incomplete data.

Result: Fail

--------------------------------------------------------------------------------------------------------------------------------------

1. Test case : Validate Add Employee With Duplicate Employee ID

Bug Description:- The form was submitted without validation error.

Pre condition :- 1.You are logged in as an Admin user in OrangeHRM.

2.At least one existing employee record already uses the Employee ID 0012.

3.You have navigated to the PIM > Add Employee page.

Test Data :- First Name: Sarah

Last Name: Connor

Employee ID: 0012 (must already exist in the system)

Severity : High

Priority : High

Expected result :- 1.System should prevent submission.

2.An error message appears such as:

3.“Employee ID already exists”

4.Employee is not created, and the user remains on the page.

Actual result :- Actual Result – Pass:

When 0012 was entered, system immediately flagged the ID as duplicate.

An error message appeared: “Employee ID already exists.”

Employee creation was blocked.

Result: Pass

Actual Result – Fail (if happened):

The form was submitted without validation error.

A second employee record with ID 0012 was created, causing duplicate primary key or data integrity issues.

Result: Fail

--------------------------------------------------------------------------------------------------------------------------------------

1. Test case : Validate Cancel Add employee form Ensure ,cancel button clears the form or navigates back

Bug Description:- Cancel button either saves data unintentionally or does not navigate back

Pre condition :- 1.User is logged in as Admin.

2.User is on the PIM > Add Employee page.

3.No data has been saved yet.

Test data :- 1.First Name: John

2.Last Name: Doe

3.Employee ID: (auto-generated or entered manually)

4.Any other optional fields filled (e.g., Job Title, Location).

Severity : Low

Priority : low

Expected Result :- 1.The system navigates back to the Employee List page.

2.No new employee record is created.

3.The form fields are cleared or discarded.

4.User sees the existing list without changes.

Actual Result :-

1. Pass: Cancel button navigates back without saving any data.

2.Fail: Cancel button either saves data unintentionally or does not navigate back.

--------------------------------------------------------------------------------------------------------------------------------------

1. Test case : Validate Add Employee with Long Names

Bug Description:- System crashes, data truncates silently, or unexpected errors occur.

Pre condition :- 1.User is logged in as Admin.

2.User is on the PIM > Add Employee page.

3.System allows input in the name fields (no prior restrictions preventing entry).

Test data : 1.First Name: A string with 100 characters (e.g., "AAAAAAAAA... (100 times)")

1. last Name: A string with 100 characters (e.g., "BBBBBBBBB... (100 times)")

Severity : Medium

Priority : Medium

Steps :-

1.The system either:

2.Accepts the long names and successfully creates the employee record, or

3.Shows a clear validation error indicating maximum allowed length exceeded.

4.No system crashes, UI breakage, or data truncation without warning.

Expected Result :- 1.The system either:

2.Accepts the long names and successfully creates the employee record, or

3.Shows a clear validation error indicating maximum allowed length exceeded.

4.No system crashes, UI breakage, or data truncation without warning.

Actual result :- 1.Pass: Employee added successfully or proper validation error shown.

2.Fail: System crashes, data truncates silently, or unexpected errors occur.

--------------------------------------------------------------------------------------------------------------------------------

38.Test case :- Validate Add Employee with Unicode Characters in Name

Bug Description:- Database or front-end fails to render Unicode properly.

Test Steps :- 1.Log in as Admin to the OrangeHRM system.

2.Navigate to PIM > Add Employee.

3.In the First Name field, enter: Алексей

4.In the Last Name field, enter: 王

5.Click on the Save button.

6.Observe the result on the Personal Details page or Employee List.

Pre condition :-

Admin user is logged into OrangeHRM and is on the PIM > Add Employee page.

Test data :- First Name: "Алексей" (Cyrillic)

- Last Name: "王" (Chinese character)

Severity :- Medium

Priority :- Medium

Expected Result :- 1.The system accepts the Unicode characters without error.

2.The employee profile is created successfully.

3.The Unicode characters are correctly displayed in:

4.Personal Details

5.Employee List

6.Search results

Actual result :-

Pass:

The Unicode name is saved and displayed correctly without corruption or errors.

No validation or encoding errors occur.

Fail:

Application throws an error like:

“Invalid characters”

“Unsupported format”

Name is saved as ????? or corrupted characters.

Database or front-end fails to render Unicode properly.

--------------------------------------------------------------------------------------------------------------------------------------

1. Test case :- Verify that the system enforces a minimum age restriction (e.g., 18 years) when adding a new employee.

Bug Description:- If the system allows saving or shows no validation

Pre condition :- 1.Admin is logged in and navigated to PIM > Add Employee page.

1. System must have a minimum age policy (e.g., employees must be at least 18 years old).

Test Data :- 1.First Name: Mark

2.Last Name: Smith

3.Date of Birth (DOB): 2010-01-01 (i.e., employee is under 18)

Severity :- Medium

Priority :- High

Expected Result :- 1.System should not allow saving the employee.

1. A validation error should appear: "Employee must be at least 18 years old."

Actual result :-

Pass – If the system displays the error and prevents saving.

Fail – If the system allows saving or shows no validation.

--------------------------------------------------------------------------------------------------------------------------------------

1. Test case :- Validate if the system accepts or restricts assigning a future joining date to an employee during the "Add Employee" process.

Bug Description:- Joining date is in an invalid format or exceeds date limits (e.g., 31/13/2030 or year 9999)

Test type :-  
Functional Testing + Boundary Testing

Ensures the system handles joining dates correctly according to business logic.

Test Method :-  
Manual Testing: Enter different joining dates (past, present, future) and observe system behavior.

Test Technique :-  
Boundary Value Analysis (BVA) + Equivalence Partitioning (EP)

Valid partition:  
If future dates are allowed → Accept today and future dates

If not → Accept only today or past dates

Invalid partition:  
1.Future dates (if not allowed)  
2.Past dates (if system is future-only)  
3.Invalid format / empty date

Steps :-

1.Log in to the OrangeHRM application using Admin/HR credentials.  
2.Navigate to the PIM module from the top menu.  
3.Click on the “Add Employee” option.  
[4.In](http://4.In) the First Name field, enter: Alice  
[5.In](http://5.In) the Last Name field, enter: Future  
6.(Optional) Enable the “Create Login Details” checkbox.  
Enter a Username: alice.future  
Enter a Password: Pass@123  
7.Set Status to: Enabled  
8.Locate or enable the Joining Date field (if visible in your configuration).  
9.Enter a future date in the Joining Date field (e.g., 30/12/2030).  
10.Click the Save button.

Pre condition:-

1.User is logged in with Admin/HR permissions.  
2.User has access to PIM → Add Employee.  
3.System date is before the selected joining date (i.e., joining date is in the future).  
4.All other required fields are filled with valid data.

Test Data :-

Field Value  
First Name Alice  
Last Name Future  
Joining Date 30/12/2030  
Username alice.future  
Password Pass@123  
Status Enabled

Boundary Test dates :- (Assuming today is 2025-07-26):

Test Date Description Expected Result (Depends on Rules)  
1.2025-07-25 :One day in the past :Accept  
2.2025-07-26 :Today's date :Accept  
3.2025-07-27: One day in the future :Yes or No (based on rule)  
2026-01-01 :Far future date :Yes or No

Severity :medium

Priority :medium

Expected Result :- If business rules allow, entry is saved; else, error is shown

Actual Result :- Pass :- 1.The system accepts a valid future date (e.g., tomorrow, next month, next year).  
[2.No](http://2.No) validation rule restricts future dates.  
3.Employee record is saved, and joining date is stored as entered.  
4.User sees “Successfully Saved” message.  
5.Employee is visible in Employee List with the future date.  
6.This is expected behavior in most HR systems for pre-joining or future onboarding.

Fail :-  
1.Joining date is in an invalid format or exceeds date limits (e.g., 31/13/2030 or year 9999).  
2.Custom validation is added to restrict future dates due to company policy.  
3.Backend throws a database error or validation exception due to misconfigured date fields.  
4.Browser/client-side validation prevents entering or selecting a date beyond a certain limit.  
5.User has insufficient permissions to modify the joining date field.

-----------------------------------------------------------------------------------------------------------------------------------------------------------------

1. Test case :- system behavior when the session times out during the employee creation process

Bug Description:- System allows form submission after timeout

Pre condition :- 1.User is logged into OrangeHRM with Admin/HR credentials.

2.User is on PIM → Add Employee screen.

3.Session timeout is configured in the system (e.g., 10 or 15 minutes of inactivity).

4.User leaves the form idle (no clicks or actions) for longer than the configured session timeout.

5.A valid internet connection is present.

Test data :- First Name Timeout

Last Name Test

Username timeout.test

Password Pass@123

Joining Date 30/12/2025

Severity : Medium

Priority : Medium

Expected Data :- System should:

1.Automatically log the user out OR

2.Show a session timeout message, e.g., "Your session has expired, please log in again".

3.Redirect the user to the login page.

4.Unsaved data is lost (unless autosave or form persistence is implemented).

5.No employee record is created.

Test Steps :-

1.Log in to OrangeHRM as an Admin or HR user.

2.Navigate to PIM → Add Employee.

3.Fill in some or all fields in the form (e.g., First Name, Last Name).

4.Do nothing (no mouse or keyboard activity) — wait for the session to expire.

5.After timeout, click on Save or try to navigate elsewhere.

Actual data :-

When Passes

1.System logs user out or prevents form submission with a clear timeout message.

2.No employee is saved.

3.User is redirected to login or shown a re-authentication prompt.

When Fails

1.System allows form submission after timeout and:

2.Redirects unexpectedly.

3.Shows a vague or broken error.

4.Saves the form with invalid session.

5.System crashes or freezes due to expired session but no proper handling.

------------------------------------------------------------------------------------------------------------------------------

41 Test case :- Verify that the system prevents creation of an employee using an email address already assigned to another employee.

Bug Description:- If the system allows saving or shows no error for the duplicate email.

Pre condition :- 1.At least one existing employee already has the email address: jane.doe@example.com.

1. Admin is logged in and navigated to PIM > Add Employee page.

Test Data :- 1.First Name: John

2.Last Name: Smith

3.Work Email: jane.doe@example.com (already in use)

Severity : High

Priority : High

Expected Result :- System should not allow saving the record.

A validation error message should appear:

"Email already in use"

The form should not be submitted.

Actual result :- Pass – If the system correctly blocks submission and shows the validation error.

Fail – If the system allows saving or shows no error for the duplicate email.

Actual result needs to be filled after test execution.

-----------------------------------------------------------------------------------------------------------------------------------

42.Test case : Verify Employee Appears in ohrm\_employee\_search Index Table

Bug description:- No record in ohrm\_employee\_search.

Pre condition:1.Admin/HR user is logged into OrangeHRM.

2.Employee John Doe has already been added successfully through the Add Employee form.

3.Optional: A job title was selected during employee creation.

4.Database access is available.

Test data : First Name John

Last Name Doe

Full Name John Doe

Job Title e.g., QA Engineer (if assigned)

Severity :- Medium — Affects search/filter results

Priority :- High — Impacts daily user functionality (search)

1.Log in to OrangeHRM as Admin or HR.

2.Navigate to PIM → Add Employee.

3.Create employee:

First Name: John

Last Name: Doe

4.Assign a job title (e.g., QA Engineer) if required.

5.Click Save.

6.After successful save, get the emp\_number:

SELECT emp\_number FROM hs\_hr\_employee WHERE emp\_firstname = 'John' AND emp\_lastname = 'Doe';

7.Use the retrieved emp\_number to query the search index table:

SELECT \* FROM ohrm\_employee\_search WHERE emp\_number = '<emp\_number>';

8.Verify the record includes:

full\_name = 'John Doe'

job\_title is correct (if assigned)

Expected Result :- 1.Record exists in ohrm\_employee\_search.

2.Fields like emp\_number, full\_name, and job\_title (if assigned) are correct.

3.Employee is visible in UI search results under PIM → Employee List.

Actual result :- PASS —

SQL query returned one record.

emp\_number, full\_name = John Doe, and job\_title match input.

Employee visible in Employee List in UI search.

Actual: PASS

FAIL if:

No record in ohrm\_employee\_search.

full\_name is incorrect.

Record present but job\_title is missing (if assigned).

Employee does not appear in UI search.

--------------------------------------------------------------------------------------------------------------------------------------

43.Test case : Validate Ensure optional fields are stored as NULL or default values when left blank during employee creation

Bug Description:- emp\_middle\_name or emp\_gender has unintended data like '-', 'Unknown', or incorrect defaults.

Pre condition :- 1.Admin/HR user is logged into OrangeHRM.

2.Employee John Doe has been added via PIM → Add Employee.

3.The following fields were left blank or unselected during creation:

Middle Name

Gender

1. Database access is available to run SQL queries.

Test data :- Field Value

First Name John

Last Name Doe

Middle Name (left blank)

Gender (not selected)

Severity :- Medium – Affects data cleanliness and reporting accuracy, but not critical to core functionality.

Priority :- Medium – Important for clean database records, especially when used in integrations or reports.

Test steps :-

1.Log in to OrangeHRM as Admin.

2.Go to PIM → Add Employee.

3.Enter the following:

First Name: John

Last Name: Doe

Leave Middle Name blank.

Do not select any Gender.

Click Save and verify the employee is saved successfully.

4.Open the database connection.

5.Run the following SQL query:

SELECT emp\_middle\_name, emp\_gender

FROM hs\_hr\_employee

WHERE emp\_firstname = 'John' AND emp\_lastname = 'Doe';

Check the values of emp\_middle\_name and emp\_gender.

Expected Result :- 1.emp\_middle\_name should be NULL (or empty string, based on DB design).

2.emp\_gender should be NULL or a system-defined default value (e.g., 0 or empty string).

3.No invalid or placeholder values (e.g., "N/A", "Unknown") should be inserted.

Actual result :- PASS –

Query returned emp\_middle\_name = NULL and emp\_gender = NULL (or proper default).

No unexpected or hardcoded values were found.

FAIL –

emp\_middle\_name or emp\_gender has unintended data like '-', 'Unknown', or incorrect defaults.

--------------------------------------------------------------------------------------------------------------------------------------

44.Testcase : Validate Add employee with missing mandatory fields

Bug Description:- Column 'FirstName' cannot be null

Pre condition :- Employee table exists

Test Data :- FirstName = NULL

LastName = "Smith"

Email = ["smith@org.com"](mailto:\"smith@org.com\")

Severity :- High

Priority :- High

Steps :- Try to insert employee with missing FirstName:

INSERT INTO Employee (FirstName, LastName, Email, Department, DateOfJoining) VALUES (NULL, 'Smith', 'smith@org.com', 'Finance', '2025-06-01');

Expected result :- The system should reject the insertion and return an error:

❗️ERROR: Column 'FirstName' cannot be null

Actual Result :- ✅ Pass – The system correctly throws a NOT NULL constraint violation error:

ERROR 1048 (23000): Column 'FirstName' cannot be null

--------------------------------------------------------------------------------------------------------------------------------------

1. Test case : Validate Add employee with invalid email format

Bug Description:- Invalid email format

Pre condition :- Employee table exists with a schema that defines Email as VARCHAR(100) NOT NULL UNIQUE.

Note: SQL itself does not validate email format unless there’s a check constraint, trigger, or validation at the application/API level.

Test data :- Field Value

FirstName John

LastName Doe

Email john.email.com ❌ (Invalid format: missing "@")

Department IT

DateOfJoining 2025-06-01

Severity :- Medium

Invalid emails can prevent future communication or cause data issues.

Priority :- High

Email validation should ideally happen both at the UI/API layer and in the DB to ensure integrity.

Expected Result :-

The system should reject the record due to invalid email format, only if validation exists at:

Database level (CHECK constraint),

Trigger level, or

Application/API level.

Actual result :-

If no format validation exists:

❗ FAIL – The record is inserted successfully, even though the email is in an invalid format.

FAIL – Unless explicitly handled by a CHECK constraint or validation logic, SQL does not reject invalid email formats.

If email format validation is implemented:

✅ PASS – The system throws an error like:

ERROR: Invalid email format

--------------------------------------------------------------------------------------------------------------------------------------

1. Test Case :- Validate Prevent Creation of Duplicate Report Name in PIM Reports

Bug Description:- No validation message is displayed. The duplicate report is saved successfully.

### Test Technique: 1.Black-box Testing Test from user’s perspective by entering report names and checking system response without accessing source code.

1.Test Type:  
1.Functional Testing  
Verify the system enforces uniqueness of report names as per the requirement.

2.Negative Testing  
Intentionally input duplicate names to check if system rejects them.

Test Method:  
Manual Testing  
Create a report, then attempt to create another with the same name, verify error message and rejection.

Steps :- 1.Navigate to PIM > Reports.

2.Click “Add”.

3.Enter "Employee Job Details" as the report name (same as existing one).

4.Click “Save”.

Pre condition:-

1.The report "Employee Job Details" must already exist in the system.

2.User must be logged in with admin or report-creation rights.

3.Access to the PIM > Reports module is available.

Test data :-

Report Name: Employee Job Details (same as existing report)

Severity :- High – Duplicate entries can lead to confusion and reporting inconsistencies.

Priority :- High – This is a core validation that should prevent duplicate data.

Exception:- An error or validation message should be displayed: “Report name already exists.” Report should not be saved.

Actual :- No validation message is displayed. The duplicate report is saved successfully.

----------------------------------------------------------------------------------------------------------------

1. Validate Mandatory Field: Report Name

Bug Description:- No validation message appears. The form submits and a blank-named report is created or error handling fails.

Test Techniques :-

1)Equivalence Partitioning  
Valid input: Alphanumeric names ("Employee Report", "Report123")

Invalid input: Blank, whitespace-only, special characters (if restricted)

Boundary Value Analysis  
Minimum characters (e.g., 1 character)

Maximum characters allowed (e.g., 100 characters)

One more than the allowed limit (should fail)

Test Method :-

Black Box Testing  
Focus on the behavior without knowing internal logic: enter/omit values and verify system response

Manual Testing  
Test field interactions and validation messages in the UI

Test Types

Functional Testing  
Ensure the "Report Name" field must be filled before submission.

1.Error message appears when field is empty

2.Usability Testing  
Ensure error messages are clear, user-friendly, and positioned properly

3.Negative Testing  
Submit the form with an empty field or invalid data and check system handling

Steps :-

1.Navigate to PIM > Reports.

2.Click “Add” to create a new report.

3.Leave the Report Name field empty.

4.Click “Save”.

Pre condition:-

1.User must be logged in with permission to create reports.

2.Access to PIM > Reports module is available.

3.No input should be provided in the Report Name field during test execution.

Test data :-

Report Name : Leave this field Blank

Severity :- High – Allows creation of invalid/unnamed records, which can cause data integrity issues.

Priority :- High – Basic validation must be enforced to prevent broken or unusable data.

Expected Result :-

System should display a validation error message: “Report name is required.”  
The report should not be created.

Actual Result:- No validation message appears. The form submits and a blank-named report is created or error handling fails.

1. Validate Dynamic Multi-Criteria Filtering in Reports

Bug Description:- Invalid filters are accepted without validation/error messages

Test Techniques:-

1.Equivalence Partitioning  
Valid combinations (e.g., HR + Active + Manager)

Invalid combinations (e.g., IT + Terminated + Intern = 0 results)

2.Boundary Value Analysis  
Max/min filters applied (e.g., 1 filter vs all possible filters)

3. Combinatorial Testing  
Permute combinations of multiple filters to verify cross-logic

e.g., Department x Job Title x Status

4.Error Guessing  
Incompatible or conflicting filter values

Special characters, blank values, or long inputs in text filters

Test Types :-  
What to Validate  
1.Functional Testing Filters work individually and in combination  
2. Usability Testing Users can easily add/remove filters dynamically  
3. Performance Testing Response time with many filters and large datasets  
4. Regression Testing New filters don’t break old filter logic or saved reports

Steps :-

1.Navigate to PIM > Reports.

2.Click Add to create a new report.

3.Enter the report name as above.

4.Add filter criteria dynamically:  
 • Job Title = "QA Engineer"  
 • AND Department = "IT"  
 • AND Years of Service >= 2

5.Select display fields: Employee ID, Full Name, Department, Years of Service.

6.Click Save.

7.Generate the report.

Verify that only employees matching all filter criteria appear.

Pre condition:-

1.Multiple reports exist with diverse employee data.

2.User has permission to create and generate reports.

3.The report creation UI supports adding multiple filter criteria dynamically (AND/OR conditions).

Test data :-

Report Name: "QA Engineers in IT Department with 2+ Years"  
Filters:

Job Title = "QA Engineer"

Department = "IT"

Years of Service >= 2

Severity :- High — tests complex filtering logic that impacts accuracy of reports.

Priority :- High — critical for users needing targeted data extraction and analysis.

Expected Result :- The generated report only contains employees who:  
• Have the Job Title "QA Engineer",  
• Work in the "IT" department,  
• Have at least 2 years of service.  
All selected fields display correctly.  
The system allows flexible combinations without errors or performance lag.

Actual Result :- Pass:

**Pass** if **all** the following conditions are met:

**Filter Functionality (Functional Testing):**

The report returns **only** employees matching **all** three filter criteria:

Job Title = "QA Engineer"

Department = "IT"

Years of Service ≥ 2

**Field Validation:**

Displayed fields include only:

Employee ID

Full Name

Department

Years of Service

**UI Behavior (Usability Testing):**

1.Filters can be dynamically added or removed without errors.

2.Filters are applied in correct logical order (AND condition as expected).

3.UI updates in real-time or on report generation without misalignment or glitches.

**Regression Testing:**

1.Older saved reports and filter combinations still work without errors.

2.New filter logic doesn’t affect pre-existing report templates or logic.

### Fail:- **Fail Criteria**:

The test **fails** if **any** of the following occur:

**Incorrect Filtering**:

1.Report includes employees who **don’t meet all filter criteria**.

2.Missing employees who do meet the criteria.

**UI Failures**:

1.Filters cannot be added/removed dynamically.

2.Invalid filters are accepted without validation/error messages.

**Field Mismatch**:

1.Wrong fields are shown in the report.

2.Missing fields from the selected display list.

**System Errors**:

1.Application crashes or shows error when filters are applied.

2.Report generation times out or is excessively delayed (>10 seconds under normal load).

**Validation Gaps**:

1.Special characters or long text in filters cause unexpected behavior.

2.Blank filters are accepted and cause logic failure.

**Saved Reports Broken**:

Existing saved reports no longer generate correctly after new filter logic is added.

--------------------------------------------------------------------------------------------------------------------------------------

1. Test case : Validate Auto-Save and Draft Recovery in Report Creation

Bug Description:- After closing the browser and logging back in, the system did not prompt to recover the unsaved report. All previously entered data (report name, selected fields, filters) was lost. The report builder opened with a blank form.

Validate Auto-Save and Draft Recovery in Report Creation

Test Techniques:-

1.State Transition Testing  
Simulate transitions: Editing → Browser Crash → Reopen Builder

2.Error Guessing  
1)Unexpected logout

2)Tab/browser closed

3)Intermittent internet connection

4)System timeout

3.Boundary Value Analysis  
Drafts with minimum and maximum field lengths

Drafts saved after 1 sec vs. after long sessions  
Test Type :Validation Focus

1.Functional Testing: Auto-save logic, recovery of drafts

2.Usability Testing :User feedback (e.g., “Draft saved”, “Recover draft?” prompt)

3.Negative Testing :System gracefully handles corrupted or partial draft data

4.Regression Testing :Auto-save and recovery still work after updates to report builder

Steps :-

1.Navigate to PIM > Reports.

2.Click Add to create a report.

3.Enter partial data (report name, one filter).

4.Wait or trigger auto-save.

5.Close the browser abruptly or log out.

6.Log back in.

7.Re-open the Reports module.

8.Check if system prompts to recover the unsaved report OR auto-restores the draft.

Pre condition:-

User is logged in with report creation permissions.

System has auto-save or draft support in report builder.

Session recovery settings are enabled.

Test Data :-

Report Name: “Attrition Trends Q4”  
Fields: Employee ID, Name, Termination Date  
Filter: Termination Date > 01-Oct-2024

Severity :- Medium – Impacts user efficiency, especially for large or complex reports.

Priority :- Medium to High – Valuable UX enhancement, especially in enterprise settings.

Expected Result :- • System should retain the unsaved data (report name and any added filters/fields).  
• Prompt to recover draft should appear OR draft auto-restored on re-entry.  
• No data loss or duplicate drafts.

Actual result :- Pass:After re-logging into the system and navigating back to the Reports module, the user was prompted to recover an unsaved report. The draft titled “Attrition Trends Q4” reloaded with all previously entered fields and filters (Employee ID, Name, Termination Date, filter on Termination Date > 01-Oct-2024).

Fail : After closing the browser and logging back in, the system did not prompt to recover the unsaved report. All previously entered data (report name, selected fields, filters) was lost. The report builder opened with a blank form.

--------------------------------------------------------------------------------------------------------------------------------------

1. Test case :Validate User is logged in with access to report generation.

Bug Description:- The system allowed report generation with a future date filter. Report generated with empty or incorrect data, causing confusion. No validation error was shown.

Test Technique :-  
1.Role-Based Testing  
Validate different user roles (Admin, HR, Employee, Guest).

2.Access Control Testing  
Check that backend endpoints require a valid token/session and proper roles.

3.Negative Testing  
Try to access report generation without logging in or with insufficient privileges

Test Types :-

1.Functional Testing Validate access control to reporting

2.Security Testing Prevent unauthorized use via URL/API

3.Regression Testing Verify continued access after updates

4.Usability Testing Ensure clear messaging for access denied

Steps :-

1.Navigate to PIM > Reports.  
2.Open an existing report or create a new report.  
3.Set filter “Termination Date” to 01-Jan-2050 (future date).  
4.Click “View” to generate the report.

Pre condition:-

1.User is logged in with access to report generation.  
2.System implements validation to restrict future dates in date filter fields (e.g., joining date, termination date).

Test data :-

Filter: Termination Date = 01-Jan-2050 (a future date)

Severity :- Medium – Future data might not exist and cause confusion or inaccurate reports.

Priority :- medium

Expected Result :- The system displays a validation error message: “Date cannot be in the future.”

Report generation is blocked until the filter is corrected.

Actual Result :- Pass :- The system displayed a validation error message: “Date cannot be in the future.” Report generation was blocked as expected. No report was generated with invalid filter criteria.

Fail :- The system allowed report generation with a future date filter. Report generated with empty or incorrect data, causing confusion. No validation error was shown.

----------------------------------------------------------------------------------------------------------------

1. Test case:-Validate Performance of Report Query with Large Data Set

Bug description:- Data is incorrectly displayed or duplicated.

Test Technique :-  
Benchmark Testing Compare performance over multiple data sizes

UI/UX :-  
Behavior :Expectation  
1.While report loads: Show loading spinner or estimated time  
2.If query takes too long :Show “This may take a while…” message  
3.If fails :Friendly error: “Query took too long or returned too much data.”  
4.Option: Let user export directly, skip preview

Steps :-

1.Log in as a valid user  
2.Navigate to PIM > Reports  
3.Open “All Active Employees” report  
4.Click View  
5.Measure time to load the full dataset

Pre condition:-

1.Database contains 10,000+ employee records  
2.User has access to a report that displays all employees

Test data :- Report: “All Active Employees” (No filter or a light filter)

Severity :medium

Priority :- medium

Expected Result :-

1.SQL query executes and returns results within acceptable threshold (e.g., < 5 seconds)  
[2.No](http://2.No) timeout, crashes, or unresponsiveness

Actual Result :-

Report generated in 3.2 seconds. Query optimized.

Pass:- 1.**No System or Application Errors**

No timeouts, crashes, or HTTP/server errors during or after report generation.

2.**UI Responsiveness**

The user interface remains responsive throughout and after report execution (no freezing, slow scrolling, or lag).

3.**Complete and Accurate Data**

1)All expected records are returned and correctly displayed.

2)No missing or truncated data rows or fields.

Fail:- 1) **System Errors**

1.Report generation results in errors such as:

1)HTTP 500 / 504 error

2)Database timeout

3)Application crash

**2)UI Freezes or Crashes**

The browser or application becomes unresponsive or needs to be forcefully closed.

1. **Incomplete or Inaccurate Output**

1.Records are missing or partially loaded.

2.Data is incorrectly displayed or duplicated.

----------------------------------------------------------------------------------------------

52 Test case : Validate Aggregated Report Output (GROUP BY)

Validate Aggregated Report Output (GROUP BY)  
Test Technique :-  
Technique :Purpose

1.Functional Testing :Validate correctness of aggregates  
2.Data Reconciliation Compare report output with SQL query

Boundary Testing Handle empty groups, single-item groups

Regression Testing Ensure no regressions on grouped logic

Test Type :-  
1.Backend Testing Validate SQL logic and output  
2.UI Testing Ensure groupings are presented clearly

3.Data Validation Check totals/averages match DB

Report Export Testing Aggregated values exported correctly  
SELECT  
d.department\_name,  
COUNT(e.employee\_id) AS employee\_count,  
AVG(e.salary) AS avg\_salary  
FROM  
employees e  
JOIN  
departments d ON e.department\_id = d.department\_id  
GROUP BY  
d.department\_name  
ORDER BY  
d.department\_name;

✅ Expected Output:  
department\_name :employee\_count: avg\_salary  
HR 5 55,000  
IT 7 70,000

Steps :-

1.Create a report: "Employee Count by Department"  
2.Select fields:  
Department Name  
Count(Employee ID)  
Generate the report  
Run equivalent SQL:

SELECT dept\_name, COUNT(emp\_id)   
FROM employees   
JOIN departments ON employees.dept\_id = departments.dept\_id   
GROUP BY dept\_name;

Match results with the report

Pre condition:-

1.A report aggregates employee counts by department  
2.GROUP BY logic supported in report builder

Test data :-

Count of employees per department

Severity :medium

Priority :medium

Expected Result :-

1.Aggregated counts match backend SQL  
[2.No](http://2.No) duplicates or miscounts

Actual result :-

Pass:

The test is considered **Passed** if **all** of the following conditions are met:

**1.Correct Aggregation**:

The report correctly groups data by Department Name (or chosen grouping field).

Aggregate functions (e.g., COUNT, AVG) return accurate values.

**2.Match with SQL Query**:

Report results match exactly with output from the backend SQL query:

SELECT

department\_name,

COUNT(employee\_id),

AVG(salary)FROM employeesJOIN departments ON employees.department\_id = departments.department\_idGROUP BY department\_name;

**3.No Duplicates or Miscounts**:

Each department appears only once in the report.

No inflated or deflated totals.

**4.UI Display is Clear**:

Grouped data is visually distinct and formatted correctly in the report (e.g., sorted by department name).

**5.Edge Cases Handled**:

Empty departments (with no employees) are either excluded appropriately or shown as 0 — per business rules.

Single-employee departments are correctly counted.

**6.Export Accuracy**:

If the report is exported (PDF/Excel), group-wise values and formats are preserved correctly.

--------------------------------------------------------------------------------------------------------------------------------------

PIM > Reports

53.Test case :- Validate Prevent Creation of Duplicate Report Name in PIM Reports

Bug Description:- No validation message is displayed. The duplicate report is saved successfully.

Pre condition :- 1. The report "Employee Job Details" must already exist in the system.

2. User must be logged in with admin or report-creation rights.

3. Access to the PIM > Reports module is available.

Test Data :- Report Name: Employee Job Details (same as existing report)

Severity :- High – Duplicate entries can lead to confusion and reporting inconsistencies.

Priority :- High – This is a core validation that should prevent duplicate data.

Test Steps :-

1. Navigate to PIM > Reports.

2. Click “Add”.

3. Enter "Employee Job Details" as the report name (same as existing one).

4. Click “Save”.

Expected Result :- An error or validation message should be displayed: “Report name already exists.” Report should not be saved.

Actual result :- No validation message is displayed. The duplicate report is saved successfully.

------------------------------------------------------------------------------------------------------------------------------------------------

1. Test case :- Validate Mandatory Field: Report Name

Bug Description:- No validation message appears. The form submits and a blank-named report is created or error handling fails.

Test steps :- 1. Navigate to PIM > Reports.

2. Click “Add” to create a new report.

3. Leave the Report Name field empty.

4. Click “Save”.

Pre condition :- 1. User must be logged in with permission to create reports.

2. Access to PIM > Reports module is available.

3. No input should be provided in the Report Name field during test execution.

Test data :- Report Name : Leave this field Blank

Severity :- High – Allows creation of invalid/unnamed records, which can cause data integrity issues.

Priority :- High – Allows creation of invalid/unnamed records, which can cause data integrity issues.

Expected Result :- System should display a validation error message: “Report name is required.”

The report should not be created.

Actual result :- No validation message appears. The form submits and a blank-named report is created or error handling fails.

------------------------------------------------------------------------------------------------------------------------------------------------

55.Test case : Validate Dynamic Multi-Criteria Filtering in Reports

Bug Description:-Incorrect Error Message:

1)A vague or unrelated message is shown (e.g., “System error” instead of “Report name already exists”).

Pre condition :-

1. Multiple reports exist with diverse employee data.

2. User has permission to create and generate reports.

3. The report creation UI supports adding multiple filter criteria dynamically (AND/OR conditions).

Test data :- Report Name: "QA Engineers in IT Department with 2+ Years"

Filters:

- Job Title = "QA Engineer"

- Department = "IT"

- Years of Service >= 2

Severity:- High — tests complex filtering logic that impacts accuracy of reports.

Priority :- High — critical for users needing targeted data extraction and analysis.

Expected Result :- The generated report only contains employees who:

• Have the Job Title "QA Engineer",

• Work in the "IT" department,

• Have at least 2 years of service.

All selected fields display correctly.

The system allows flexible combinations without errors or performance lag.

Actual result :-Pass Criteria

1.The test is considered a PASS if all of the following are true:

2.Duplicate Detection Works:

1)When a user attempts to save a report with a name that already exists, the system detects the duplicate.

3.Proper Error Handling:

1)A clear error or validation message is shown, e.g.:

2)“Report name already exists.”

4.Report Is Not Saved:

1)The duplicate report is not saved in the system.

2)User remains on the form or is prompted to enter a unique name.

5.Case Insensitivity (if applicable):

1)If “Employee Job Details” and “employee job details” are treated as duplicates, system blocks it accordingly.

6.Consistent Behavior Across Modules:

1)Validation behaves the same in all relevant workflows (create, clone, import, etc.).

Fail Criteria:-

1.The test is considered a FAIL if any of the following occur:

2.No Validation Message Displayed:

1)The system allows saving a report with an existing name without any warning or message.

2)Duplicate Report Is Saved:

3)The system creates two reports with the same name, leading to confusion and potential data integrity issues.

3.Incorrect Error Message:

1)A vague or unrelated message is shown (e.g., “System error” instead of “Report name already exists”).

4.Validation Bypass:

1)The user can bypass the restriction via minor changes (e.g., adding a space at the end, using mixed cases) when that’s not allowed per business rules.

5.Inconsistent Validation:

1)UI allows it but backend/API later fails, or vice versa.

---------------------------------------------------------------------------------------------------------------------------------------------

----------------------------------------------------------------------------------------------------------------------------------------------

56.Test case :- Validate Auto-Save and Draft Recovery in Report Creation

Bug Description:- After closing the browser and logging back in, the system did not prompt to recover the unsaved report. All previously entered data (report name, selected fields, filters) was lost. The report builder opened with a blank form.

Pre condition :- 1. User is logged in with report creation permissions.

2. System has auto-save or draft support in report builder.

3. Session recovery settings are enabled.

Test Data :- Report Name: “Attrition Trends Q4”

Fields: Employee ID, Name, Termination Date

Filter: Termination Date > 01-Oct-2024

Severity :- Medium – Impacts user efficiency, especially for large or complex reports.

Priority :- Medium to High – Valuable UX enhancement, especially in enterprise settings.

Expected Result :- • System should retain the unsaved data (report name and any added filters/fields).

• Prompt to recover draft should appear OR draft auto-restored on re-entry.

• No data loss or duplicate drafts.

Actual result :- Pass:After re-logging into the system and navigating back to the Reports module, the user was prompted to recover an unsaved report. The draft titled “Attrition Trends Q4” reloaded with all previously entered fields and filters (Employee ID, Name, Termination Date, filter on Termination Date > 01-Oct-2024).

Fail : After closing the browser and logging back in, the system did not prompt to recover the unsaved report. All previously entered data (report name, selected fields, filters) was lost. The report builder opened with a blank form.

------------------------------------------------------------------------------------------------------------------------------------------------

57.Test case : Validate User is logged in with access to report generation.

Bug Description:- The system allowed report generation with a future date filter. Report generated with empty or incorrect data, causing confusion. No validation error was shown.

System implements validation to restrict future dates in date filter fields (e.g., joining date, termination date).

Pre condition :- 1.User is logged in with access to report generation.

1. System implements validation to restrict future dates in date filter fields (e.g., joining date, termination date).

Test data :- Filter: Termination Date = 01-Jan-2050 (a future date)

Severity :- Medium – Future data might not exist and cause confusion or inaccurate reports.

Priority- Medium

Test steps :- 1.Navigate to PIM > Reports.

2.Open an existing report or create a new report.

3.Set filter “Termination Date” to 01-Jan-2050 (future date).

4.Click “View” to generate the report.

Expected Result :- The system displays a validation error message: “Date cannot be in the future.”

Report generation is blocked until the filter is corrected.

Actual result :- Pass :- The system displayed a validation error message: “Date cannot be in the future.” Report generation was blocked as expected. No report was generated with invalid filter criteria

Fail :- The system allowed report generation with a future date filter. Report generated with empty or incorrect data, causing confusion. No validation error was shown.

------------------------------------------------------------------------------------------------------------------------------------------------

58.Test case :- Validate Performance of Report Query with Large Data Set

Bug Description:- **UI Freezes or Crashes**

The browser or application becomes unresponsive or needs to be forcefully closed.

Test Condition :-

1.Database contains 10,000+ employee records

2.User has access to a report that displays all employees

Test data :- Report: “All Active Employees” (No filter or a light filter)

Severity :- Medium

Prirority : medium

Test steps :- 1.Log in as a valid user

2.Navigate to PIM > Reports

3.Open “All Active Employees” report

4.Click View

5.Measure time to load the full dataset

Expected Result :- 1.SQL query executes and returns results within acceptable threshold (e.g., < 5 seconds)

1. No timeout, crashes, or unresponsiveness

Actual Result :- Report generated in 3.2 seconds. Query optimized.

Report generated in 3.2 seconds. Query optimized.

Pass:

1.No System or Application Errors

2.No timeouts, crashes, or HTTP/server errors during or after report generation.

3.UI Responsiveness

4.The user interface remains responsive throughout and after report execution (no freezing, slow scrolling, or lag).

5.Complete and Accurate Data

6.All expected records are returned and correctly displayed.

7.No missing or truncated data rows or fields.

8.Stable Resource Usage

9.CPU and memory usage remain within expected thresholds (optional, if monitored).

10.No memory leaks or browser crashes.

Fail:-

11.Exceeds Performance Threshold

12Report generation takes longer than the expected limit (e.g., > 5–10 seconds under standard load).

13System Errors

14Report generation results in errors such as:

15HTTP 500 / 504 error

16Database timeout

17Application crash

18UI Freezes or Crashes

19The browser or application becomes unresponsive or needs to be forcefully closed.

20Incomplete or Inaccurate Output

21Records are missing or partially loaded.

22Data is incorrectly displayed or duplicated.

23High Resource Consumption

24Memory usage spikes drastically (e.g., >80% of browser/system capacity).

25System performance is impacted after report generation.

------------------------------------------------------------------------------------------------------------------------------------------------

59.Test case :- Validate Aggregated Report Output (GROUP BY)

Bug Description:- Records are missing or partially loaded.

Pre condition :-

1.A report aggregates employee counts by department

2.GROUP BY logic supported in report builder

Test data :- Count of employees per department

Severity :- medium

Prirority :- medium

Test steps :-

1.Create a report: "Employee Count by Department"

2.Select fields:

Department Name

Count(Employee ID)

Generate the report

Run equivalent SQL:

SELECT dept\_name, COUNT(emp\_id)

FROM employees

JOIN departments ON employees.dept\_id = departments.dept\_id

GROUP BY dept\_name;

Match results with the report

Expected result :- 1.Aggregated counts match backend SQL

1. No duplicates or miscounts

Actual Result :- 1.The test is considered Passed if all of the following conditions are met:

2.Correct Aggregation:

3.The report correctly groups data by Department Name (or chosen grouping field).

4.Aggregate functions (e.g., COUNT, AVG) return accurate values.

5.Match with SQL Query:

6.Report results match exactly with output from the backend SQL query:

SELECT

department\_name,

COUNT(employee\_id),

AVG(salary)

FROM employees

JOIN departments ON employees.department\_id = departments.department\_id

GROUP BY department\_name;

7.No Duplicates or Miscounts:

8.Each department appears only once in the report.

9.No inflated or deflated totals.

10.UI Display is Clear:

11.Grouped data is visually distinct and formatted correctly in the report (e.g., sorted by department name).

12.Edge Cases Handled:

13.Empty departments (with no employees) are either excluded appropriately or shown as 0 — per business rules.

14.Single-employee departments are correctly counted.

15.Export Accuracy:

16.If the report is exported (PDF/Excel), group-wise values and formats are preserved correctly.

Fail Criteria:

1.The test is Failed if any of the following occur:

2.Incorrect Aggregates:

3.Counts or averages do not match SQL query results.

4.Values are incorrect, missing, or inflated.

5.Grouping Logic Error:

6.Departments are not properly grouped (e.g., multiple rows for same department).

7.Some groups are skipped or misclassified.

8.UI Issues:

9.Aggregated data not clearly formatted or aligned.

10.Values appear misaligned or nested improperly.

11Mismatch with SQL Output:

12Report output differs from backend SQL query results.

13Data Export Fails:

14Exported report shows incorrect or inconsistent group totals compared to the on-screen display.

15Regression Issues:

16If previously working GROUP BY reports are now failing or returning inaccurate results after a new deployment.

------------------------------------------------------------------------------------------------------------------------------------------------

60.Test case :- Validate Duplicate Entry Prevention in Reports

Bug Description:-Any employee\_id appears **more than once** in report output.

Pre condition :-

1.Employee data must be uniquely identified by employee\_id

2.Join queries used in reports shouldn't introduce duplicates

Test data :-

A report joining employee and job\_history tables (e.g., multiple job history entries)

Severity :- high

Priority :- High

Test Steps :-

1.Generate a report with fields: Employee ID, Name, Current Job Title

2.Internally validate the SQL is using only latest job entry OR deduplication logic

Run backend SQL to detect duplicates:

SELECT employee\_id, COUNT(\*)

FROM report\_output

GROUP BY employee\_id

HAVING COUNT(\*) > 1;

Cross-check with UI report

Expected Result :- 1.No duplicate employee records appear in the report

2.SQL uses correct DISTINCT, ROW\_NUMBER(), or subquery logic

#### Actual Result :- **Pass Criteria**:

1.Each employee\_id appears **only once** in the report.

2.Backend SQL is **validated to use** deduplication techniques:

DISTINCT, subqueries, ROW\_NUMBER(), or aggregate filters like MAX(start\_date) for latest job.

3.Validation query confirms **no employee\_id with** COUNT > 1 in output.

4.UI matches backend — no duplicated rows seen visually.

5.Join logic correctly restricts to one job per employee (e.g., latest).

#### **Fail Criteria**:

1.Any employee\_id appears **more than once** in report output.

2.SQL uses incorrect join or lacks deduplication (e.g., raw LEFT JOIN without filtering job history).

3.UI report shows duplicate rows.

4.Exported report has multiple identical records.

5.Data changes or schema updates reintroduce duplicates (regression).

---------------------------------------------------------------------------------

61.Test case: Verify Orphan Data Handling in Reports

Bug Description:- **Duplicate Employee Records Exist**:

The same employee\_id appears **multiple times** in the report output.

Test steps :- 1.Run above SQL to detect orphaned records

1. Check if the report filters them or handles gracefully (e.g., shows "N/A")

Pre condition:- Some employee records exist with job\_id or department\_id not mapped

Test data :- SELECT \* FROM employees

WHERE job\_id NOT IN (SELECT job\_id FROM jobs)

OR department\_id NOT IN (SELECT department\_id FROM departments);

Severity : medium

Priority : medium

Expected Result :- UI does not crash; shows “N/A” or error if orphaned reference is encountered

Actual result :-

1.**No Duplicate Records**:

Each employee\_id appears **only once** in the report output.

2.**Correct SQL Logic**:

The report query uses proper deduplication logic:

DISTINCT

ROW\_NUMBER() OVER (PARTITION BY ...)

Aggregation (e.g., MAX(start\_date) for latest job)

Proper subqueries or filters on job\_history

**Join Conditions are Accurate**:

Joins between employees and job\_history (or other tables) do **not cause row duplication**.

**Data Validation Matches**:

Cross-verification with the backend SQL query:

SELECT employee\_id, COUNT(\*)

FROM report\_output

GROUP BY employee\_id

HAVING COUNT(\*) > 1;

returns **zero rows**.

· **UI Reflects Accurate Data**

The user interface displays **no duplicate rows**.

Sorting, filtering, or exporting the report does not introduce duplicates.

· **Regression Coverage**:

No duplicate records reappear after updates to schema, filters, or report logic.

### **Fail Criteria**

The test is **considered failed** if **any** of the following are true:

**Duplicate Employee Records Exist**:

The same employee\_id appears **multiple times** in the report output.

**Incorrect SQL or Join Logic**:

Missing DISTINCT or ROW\_NUMBER() when needed.

Joins result in one-to-many relationships that are not correctly handled.

**Backend Data Mismatch**:

Validation SQL query (using GROUP BY employee\_id HAVING COUNT(\*) > 1) returns **duplicate entries**.

**UI Displays Duplicates**:

Duplicate records visible in the UI report grid or export.

**Incorrect Handling of Job History**:

The report shows **multiple job entries** for one employee when only the **latest** should appear.

**Regression Detected**:

Duplicates appear in reports that previously worked correctly, after updates or deployments.

------------------------------------------------------------------------------------------------------------------------------------

62.Test case : Validate Field Formatting (Date/Number) or

Validate Date/Salary Format in Report Matches SQL Format

Bug Description:- Date format issue.

Test steps :- 1.View the report with salary and joining date

1. Compare UI display format with formatted SQL results

Pre condition :- System should format salary (e.g., 2 decimal places), and dates (e.g., DD-MM-YYYY)

Test data :- SQL:

SELECT employee\_id, salary, DATE\_FORMAT(joining\_date, '%d-%m-%Y') AS formatted\_date

FROM employees;

Severity :- low

Priority: medium

Expected result :- UI shows correctly formatted dates and numbers

Actual result :-1.Date Formatting:

Date fields (e.g., joining\_date) are displayed in the UI in the correct format:

Expected: DD-MM-YYYY (e.g., 15-07-2025)

Format matches the backend SQL result (e.g., from DATE\_FORMAT(joining\_date, '%d-%m-%Y')).

Dates are consistently formatted across all rows.

Exported reports (CSV, PDF, Excel) preserve the same date format as displayed in the UI.

2. Salary Formatting:

Salary values are shown with exactly two decimal places (e.g., 50,000.00).

Values match the actual SQL results (no rounding/truncation issues).

Thousands separators and currency symbols (if applicable) are correctly applied.

Exported formats match UI formatting.

3. Locale/Region Support (if applicable):

When region settings are changed (e.g., US vs EU), formatting adapts accordingly:

US: 07/31/2025, 50,000.00

EU: 31.07.2025, 50.000,00

🧾 Consistency Across Views:

The formatting remains identical in:

Report preview in UI

Exported files

Paginated views

Fail Criteria:-

The test is considered a Fail if any of the following occur:

1.Date Format Issues:

Date format in UI does not match expected format (DD-MM-YYYY).

Inconsistencies in date formats across rows.

Placeholder values or formatting errors appear (e.g., 0000-00-00, null, or 2025/31/07).

Exported report shows default DB format (e.g., YYYY-MM-DD) instead of formatted output.

2. Salary Format Issues:

Salary does not display two decimal places (e.g., 50000 instead of 50,000.00).

Mismatch between UI and SQL salary values.

Missing thousands separators or currency symbols (if applicable).

Exported reports display raw values or inconsistently formatted salaries.

3. Localization Issues:

Region-specific settings are ignored or applied incorrectly.

Formats do not change based on user locale (if feature is supported).

------------------------------------------------------------------------------------------------------------------------------------

1. Test case :- Validate direct URL Access to Configuration pages URLs Directly Without Permission

Bug Description:- Redirects but with partial content visible

Pre condition :- 1.A non-admin user (e.g., HR or Employee) account is created with no access to the Configuration dropdown or its underlying pages.

2.User is logged in and authenticated in the system.

3.URLs of the configuration pages are known and testable.

Test data :- Field Value

1.Username employee\_user

2.Password employee\_pass123

3.Role Employee (restricted access)

4.Browser Google Chrome (latest version)

5.Environment QA or Staging (https://qa.hrms-example.com)

Severity :- Critical – Unauthorized access may lead to data breaches or misuse

Priority :- High – Must be fixed immediately if access control is broken

Test steps :- 1.Launch the HRMS web application.

2.Log in using the credentials of a restricted user (e.g., Employee).

3.Bypass the UI — do not use the Configuration dropdown.

4.Manually enter the following URLs in the browser one by one:

https://qa.hrms-example.com/pim/customFields

https://qa.hrms-example.com/pim/optionalFields

https://qa.hrms-example.com/pim/dataImport

5.Press Enter and observe the behavior of the application.

6.Record whether the user is redirected, blocked, or granted access.

Expected result :- For all restricted users:

1.Direct access to Configuration pages should be blocked.

2.The system should display a message like:

"You do not have permission to view this page"

"Access Denied"

Or redirect to dashboard/home page.

1. No page content (tables, buttons, forms) should be visible.

Actual result :-Pass: User blocked with “Access Denied”

Fail:-  
1.Page loads and displays restricted content — Fail (Major Security Risk)  
2. Redirects but with partial content visible — Fail  
3.Silent failure with blank screen (no feedback) — Partial Fail

---------------------------------------------------------------------------------------------------------------------------------------------------

1. Validate the mandatory System fields are not present /not editable in PIM \_configouration \_ Optional field

Bug Description:- Changes to required fields allowed and affect form — major data flow bug.

PIM > Configuration > Optional Fields

Test type :- Functional Testing

Technique :- Black box testing  
Method :- Manual  
Test level :- UI/Component

Steps :- 1.Log in to the HRMS system with Admin credentials.  
2.Go to PIM > Configuration > Optional Fields.  
3.Observe the list of fields displayed.  
4.Identify fields known to be required by the system (e.g., First Name, Last Name, Employee ID).  
Verify whether these required fields are:  
5.Displayed in the Optional Fields list.  
6.Have editable toggle/toggles.  
7.Attempt to interact with or change the checkbox state (if any) next to a required field.  
8.Compare this with truly optional fields like "Nick Name" or "Smoker".

Pre condition:-

1.User is logged in as Admin (or role with access to Optional Fields).  
2.PIM module is accessible.  
3.There are required fields in the system that are not meant to be configurable.

Test Data :-

Field Name\_Field Type\_Expected Status\_Editable  
1.First Name\_Required\_Should NOT be editable  
2.Last Name\_Required\_Should NOT be editable  
3.Employee ID\_Required\_Should NOT be editable  
4.Nick Name\_Optional\_Editable

Severity :- High – Allowing required fields to be disabled would compromise form and data integrity.

Priority :- High – Must be enforced to preserve business rules and prevent misconfiguration.

Expected Result :- 1.Required fields (e.g., First Name, Last Name, Employee ID):  
Should not appear in the Optional Fields list.  
2.If displayed, they should be grayed out or non-editable.  
[3.No](http://3.No) toggle should be present or editable for required fields.  
4. On Add Employee form, required fields are always shown regardless of optional fields settings .

Actual result :- Pass:Required fields are hidden or disabled in config  
Fail :-  
1."First Name" toggle appears and is editable — Fail (serious issue).

2.Changes to required fields allowed and affect form — major data flow bug.

---------------------------------------------------------------------------------------------------------------------------------------------------

1. Validate Optional Fields Appear on Employee Form Only When Enabled in configouration.

PIM > Configuration > Optional Fields / Employee Personal Details

Test Type :- Functional Testing,  
Configuration Testing

Technique :- Black box testing

Method :- Manual /UI based

test level :- Integration (UI + configourable module )

Steps:-

1.Log in as Admin.  
2.Navigate to PIM > Configuration > Optional Fields.  
3.Enable an optional field that is currently disabled (e.g., check Nick Name).  
4.Disable a field that is currently enabled (e.g., uncheck Smoker).  
5.Click Save configouration and ensure a success message appears.  
6.Go to PIM > Employee List.> Add Employee .( Open existing employee for editing )  
7.Select any employee and click Edit.  
Verify whether:  
8.Nick Name field now appears in the employee form.  
9.Smoker field is no longer visible in the employee form.  
10.Optionally, revert changes to restore original field configuration.

Pre condition:-

1.loggin as Admin or user has with access to both configouration and employee management.  
[2.At](http://2.At) least one employee record exists.( optional fied is available)  
3.The PIM module -> Add employee personal details forms/page are accessible.

Test Data :-

Optional Field\_Initial Status\_Change Action\_Expected on Employee Form  
1.Nick Name\_ Disabled\_Enable Field should appear  
2.Smoker Enabled\_ Disable \_Field should disappear

Field Name State in Config Expected on Form  
1.Nickname Enabled Should appear  
2.SSN Disabled Should NOT appear

Severity :- High – Incorrect visibility affects data input, user confusion, and reporting accuracy.

Priority :- High – Important for proper employee data management and UI consistency.

Expected Result :- 1.Only the enabled optional fields (e.g., Nick Name) should be visible on the employee edit form.  
2.Disabled fields (e.g., Smoker) should not appear.  
3.The form layout and save functionality should not be impacted by the visibility change.

Actual result :- Pass:-

1.Functional Behavior:

2.Configuration Changes Are Saved:

1)Enabling or disabling fields (like "Nick Name" or "Smoker") from PIM > Configuration > Optional Fields is saved successfully with a success confirmation message.

2)Employee Form Reflects Changes Immediately:

3)The enabled field (e.g., "Nick Name") becomes visible on the employee add/edit form.

4)The disabled field (e.g., "Smoker") becomes hidden on the form.

5)No Layout Issues:

The visibility change does not break or shift the layout of the employee form.

6)No Save Issues:

Saving or updating an employee record works normally with the new field configuration.

7)Reversibility Works:

Reverting the optional field settings restores the form visibility accordingly.

8)Consistency:

Changes apply to both new employee form and edit existing employee form.

Fail Criteria

1.The test is considered a FAIL if any of the following occur:

2.Configuration Issues:

1)Changes Not Saved:

After enabling/disabling fields, no success message is shown and settings are not retained.

2)Employee Form Does Not Reflect Config Changes:

An enabled field (e.g., "Nick Name") does not appear.

A disabled field (e.g., "Smoker") still appears.

3)Inconsistent Visibility:

Some employee records show the field while others do not (inconsistent behavior).

4)UI/Layout Broken:

The form layout breaks or appears misaligned after hiding/showing fields.

5)Form Errors:

Attempting to save employee data throws an error due to missing or hidden fields that the backend still expects.

6)Reversion Fails:

Reverting the field settings does not reflect back in the employee form.

------------------------------------------------------------------------------------------------------------------------------------

1. Validate the mandatory System fields are not present /not editable in PIM \_configouration \_ Optional field

Bug Description:- "First Name" toggle appears and is editable

PIM > Configuration > Optional Fields

Test type :- Functional Testing

Technique :- Black box testing

Method :- Manual

Test level :- UI/Component

Steps :-

1.Log in to the HRMS system with Admin credentials.

2.Go to PIM > Configuration > Optional Fields.

3.Observe the list of fields displayed.

4.Identify fields known to be required by the system (e.g., First Name, Last Name, Employee ID).

Verify whether these required fields are:

5.Displayed in the Optional Fields list.

6.Have editable toggle/toggles.

7.Attempt to interact with or change the checkbox state (if any) next to a required field.

8.Compare this with truly optional fields like "Nick Name" or "Smoker".

Pre condition:-

1.User is logged in as Admin (or role with access to Optional Fields).

2.PIM module is accessible.

3.There are required fields in the system that are not meant to be configurable.

Test data:-

Field Name\_Field Type\_Expected Status\_Editable

1.First Name\_Required\_Should NOT be editable

2.Last Name\_Required\_Should NOT be editable

3.Employee ID\_Required\_Should NOT be editable

4.Nick Name\_Optional\_Editable

Severity :- High – Allowing required fields to be disabled would compromise form and data integrity.

Priority :- High – Must be enforced to preserve business rules and prevent misconfiguration.

Expected Result :-

1.Required fields (e.g., First Name, Last Name, Employee ID):

Should not appear in the Optional Fields list.

2.If displayed, they should be grayed out or non-editable.

3.No toggle should be present or editable for required fields.

4. On Add Employee form, required fields are always shown regardless of optional fields settings .

Actual result :-

Pass:Required fields are hidden or disabled in config

Fail :-

1."First Name" toggle appears and is editable — Fail (serious issue).

2.Changes to required fields allowed and affect form — major data flow bug.

------------------------------------------------------------------------------------------------------------------------------------

65.Validate Optional Fields Appear on Employee Form Only When Enabled in configouration.

Bug Description:- Reversion Fails:

Reverting the field settings does not reflect back in the employee form.

PIM > Configuration > Optional Fields / Employee Personal Details

Test Type :- Functional Testing,

Configuration Testing

Technique :- Black box testing

Method :- Manual /UI based

test level :- Integration (UI + configourable module )

Test steps :-

1.Log in as Admin.

2.Navigate to PIM > Configuration > Optional Fields.

3.Enable an optional field that is currently disabled (e.g., check Nick Name).

4.Disable a field that is currently enabled (e.g., uncheck Smoker).

5.Click Save configouration and ensure a success message appears.

6.Go to PIM > Employee List.> Add Employee .( Open existing employee for editing )

7.Select any employee and click Edit.

Verify whether:

8.Nick Name field now appears in the employee form.

9.Smoker field is no longer visible in the employee form.

10.Optionally, revert changes to restore original field configuration.

Pre condition:-

1.loggin as Admin or user has with access to both configouration and employee management.

2.At least one employee record exists.( optional fied is available)

3.The PIM module -> Add employee personal details forms/page are accessible.

Test data:-

Optional Field\_Initial Status\_Change Action\_Expected on Employee Form

1.Nick Name\_ Disabled\_Enable \_Field should appear

2.Smoker\_ Enabled\_ Disable \_Field should disappear

Field Name \_State in Config \_Expected on Form

1.Nickname\_ Enabled \_Should appear

2.SSN\_ Disabled\_ Should NOT appear

Severity :- High – Incorrect visibility affects data input, user confusion, and reporting accuracy.

Priority :- High – Important for proper employee data management and UI consistency.

Expected Result :- 1.Only the enabled optional fields (e.g., Nick Name) should be visible on the employee edit form.

2.Disabled fields (e.g., Smoker) should not appear.

3.The form layout and save functionality should not be impacted by the visibility change.

Actual Result :- Pass:-

1.Functional Behavior:

2.Configuration Changes Are Saved:

1)Enabling or disabling fields (like "Nick Name" or "Smoker") from PIM > Configuration > Optional Fields is saved successfully with a success confirmation message.

2)Employee Form Reflects Changes Immediately:

3)The enabled field (e.g., "Nick Name") becomes visible on the employee add/edit form.

4)The disabled field (e.g., "Smoker") becomes hidden on the form.

5)No Layout Issues:

The visibility change does not break or shift the layout of the employee form.

6)No Save Issues:

Saving or updating an employee record works normally with the new field configuration.

7)Reversibility Works:

Reverting the optional field settings restores the form visibility accordingly.

8)Consistency:

Changes apply to both new employee form and edit existing employee form.

Fail Criteria

1.The test is considered a FAIL if any of the following occur:

2.Configuration Issues:

1)Changes Not Saved:

After enabling/disabling fields, no success message is shown and settings are not retained.

2)Employee Form Does Not Reflect Config Changes:

An enabled field (e.g., "Nick Name") does not appear.

A disabled field (e.g., "Smoker") still appears.

3)Inconsistent Visibility:

Some employee records show the field while others do not (inconsistent behavior).

4)UI/Layout Broken:

The form layout breaks or appears misaligned after hiding/showing fields.

5)Form Errors:

Attempting to save employee data throws an error due to missing or hidden fields that the backend still expects.

6)Reversion Fails:

Reverting the field settings does not reflect back in the employee form.

------------------------------------------------------------------------------------------------------------------------------------

66.Validate Configouration dropdown not opens , when Configouration javascript is disabled

Bug Description:- Some styles/layouts broken.

TC\_PIM\_CONFIG\_JS\_DISABLED\_001

Type : Negative , Accessibility

Techniques : Black box testing

Method :- Manual testing

Steps :-

1.Launch the browser.

2.Disable JavaScript (e.g., via browser settings or developer tools).

3.Log in to the application using valid admin credentials.

4.Navigate to the PIM module.

5.Try to click on the "Configuration" dropdown on the PIM page.

6.Observe the behavior of the dropdown and the UI.

Pre condition:-

1.User has access to the PIM module.

2.User is logged in as Admin or any role that can see the Configuration dropdown.

3.JavaScript is disabled in the browser.

Test data :-

Field \_Value

1.Username \_admin.user

2.Password \_Admin@123

3.Browser \_Chrome / Firefox (JS disabled manually via dev tools or extension)

Severity :-

Medium Impacts users with JS disabled but not system-critical.

Priority :- low -medium

Expected Result :- 1.The Configuration dropdown should not open.

2.The application may show a graceful message like "Some features may not work without JavaScript."

3.The page should not break or crash.

4.No unexpected console errors or blank sections.

Actual result :-

Actual Result (Example)

Pass: Dropdown does not open (as expected).

Fail: No message shown — confusing UX.

Fail: Some styles/layouts broken.

(Note: Actual result depends on your environment.)

------------------------------------------------------------------------------------------------------------------------------------

67.Validates when Configuration dropdown in the PIM module loads data dynamically, system provides appropriate user feedback (e.g., loading spinner, disabled state) under slow or delayed network conditions.

Bug Description:- Dropdown appears empty before loading completes.

TC\_PIM\_CONFIG

Test Type :- UX / Performance / Error Handling

Technique :- Black Box Testing

Method :- Manual Testing

Testing Level :- UI + Integration Level

Test steps :-

1.Open browser and log in as an admin.

2.Go to the PIM module.

3.Open Developer Tools (F12).

4.Go to the Network tab and throttle network to Slow 3G or add custom throttling.

5.Click the Configuration dropdown.

6.Observe the dropdown button and area where options load.

7.Wait for the data to finish loading.

8.Remove throttling and reload to confirm normal behavior.

Pre condition:-

1.Admin user is logged in.

2.Configuration dropdown loads options from the backend dynamically (AJAX or API call).

3.Browser supports developer tools or network throttling.

Test data :-

Field \_Value

1.Username\_ admin.user

2.Password\_ Admin@123

3.Browser\_ Chrome or Firefox (DevTools enabled)

4.Network Mode\_ Slow 3G / Custom Throttle (simulate ~2000ms latency)

Severity :- Field Value Reasoning

Severity Medium–High Affects user experience, especially on poor connections.

Priority :- Priority Medium Not a blocker, but important for good usability.

Expected Result :- 1.A loading spinner, progress indicator, or disabled dropdown button is shown until data is loaded.

2.The user is clearly informed that the dropdown is loading.

3.Once loaded, all options are displayed correctly.

4.No multiple triggers or duplicate dropdown opens.

Actual result :-

Pass:- Spinner shown and options load correctly after delay.

Fail :-

1.No feedback shown; user keeps clicking, causing multiple requests.

2. Dropdown appears empty before loading completes.

3. UI freezes or hangs during delay.

------------------------------------------------------------------------------------------------------------------------------------

68.Validates by Simulate Multi-Tab Configuration Conflict

Bug Description:- 1. Both tabs save conflicting data — and config becomes inconsistent.

PIM> Configouration

Test type :- UX

method :- Manual

Testing level :- Integration/UI + backend

Test steps :-

1.Log in as an admin in Tab 1, navigate to PIM → Configuration → Optional Fields.

2.Open a new browser tab, log in again as the same admin (Tab 2), go to the same page.

3.In Tab 1, enable a field (e.g., “SSN”) and click Save, note the time.

4.In Tab 2, disable the same field without refreshing the tab and click Save.

5.Refresh the page in Tab 1 and verify which setting is applied.

Observe whether:

6.A conflict warning is shown.

7.The last saved change overrides silently.

8.Any audit trail is updated.

9.UI reflects the correct state consistently.

Pre condition:-

1.Admin user is logged into OrangeHRM.

2.Configuration settings (e.g., Optional Fields) are editable.

3.Same user can open multiple sessions/tabs simultaneously.

Test data :-

1.Username\_ admin.user

2.Password\_ Admin@123

3.Setting A \_Enable "SSN" field

4.Setting B\_ Disable "SSN" field

Severity :- Medium–High: Could cause silent config overwrites in real-world use cases.

Priority :- Medium :Not always critical, but important for enterprise reliability.

Expected Result :- System should:

1.Prevent overwrite if data has changed in another session,

or

Apply last-save-wins logic and reflect the latest config,

or

Warn user about changes made in other sessions (ideal).

1. No data corruption, UI glitch, or inconsistent display should occur.

Actual Result :-

Pass:- Tab 2’s change overwrites Tab 1’s without issue — last-save-wins.

Fail:- 1. Both tabs save conflicting data — and config becomes inconsistent.

2.No warning or audit logged for the conflict — risk of silent loss.

-----------------------------------------------------------------------------------------------------------------------------------

69.Validate custom field page displays list of existing custom fields and shows the correct records found label.

Bug Description:- The page loads, but no custom fields are shown in the table.

PIM \_Configouration\_Custom Field

Test type :- functional

Technique :- Black box

Test level :- System

Test method :- Manual

Steps :-

1. Login to orange application as Admin user.

2.Navigate to PIM-> Configouration -> custom fields

3. Observe the table or list displayed in the page

4. check the label above or below the list showing the total number of records (1 record found)

Pre condition:-

1. User is loggedin with admin credential .

2. At least one custom field is already created under PIM->Configouration->Custom fields

3. User has permission to access and manage PIM module.

Test data :-

Field Label\_ Section\_ Screen \_Field Type

1.Employee Hobbies\_ Personal Details\_ Personal Information\_ Text

2.Emergency Contact Number \_Contact Details \_Contact Information\_ Number

Severity :Low

Priority :medium

Expected Result :-

1. Custom fields appear correctly in the grid /table

2.A label such as 1 record is shown.

3. Filed name,section,type, screen (personal details) are all visible and accurate .

Actual result :-

Pass :-

1.The page loads successfully.

2.The custom fields table displays Employee Hobbies and Emergency Contact Number.

3.The label at the bottom reads: "2 Records Found".

4.All field details (Section, Screen, Type) appear correctly.

Fail :-

1.The page loads, but no custom fields are shown in the table.

2.The label shows: "0 Records Found", even though data exists in the system.

------------------------------------------------------------------------------------------------------------------------------------

70.Validate record found count decreases appropriately when custom field is deleted .

Bug Dscription:- Page is blank without message ,No message or table shown.

PIM\_Customfield

Type :- Functional

Technique :- black box

Level :- System

Method :- Manual

Steps:-

1.Log in to the OrangeHRM system as an Admin user.

2.Navigate to PIM → Configuration → Custom Fields.

3.Confirm the current count shown (e.g., “2 Records Found”).

4.Identify and click Delete icon next to one of the fields.

5.Confirm the deletion when prompted.

6.Wait for the page to reload (or manually refresh if needed).

7.Observe the new count message (e.g., “1 Record Found”).

Pre condition:-

1. User logged in as admin

2.There are atleast 2 custom fields present in the system.

Test data :-

Field Label\_ Screen\_ Type\_ Section

1.Emergency Contact \_Personal Details \_Text\_ Contact Information

2.Alternate ID\_ Job Details\_ Number\_ Identification

Severity :medium

Priority :medium

Expected Result:- 1.The selected custom field is removed from the list.

2.The “Record(s) Found” label decreases by one (e.g., from “2 Records Found” to “1 Record Found”).

3.No errors or broken UI elements are present.

4.The correct remaining custom field(s) should still be visible.

Actual result :-

Pass :- Field deleted and count updated to 1 Record Found.

Fail:- Field deleted but count still shows 2

------------------------------------------------------------------------------------------------------------------------------------

71.Validate that custom field with data type "Text (Single Line)" can be created and accepts valid alphanumeric input within the allowed character limit.

Bug Description:- Field should display validation if input is empty (if required).

PIM\_Config\_CustomField\_Text

Test type :- Functional

Test method :- manual

Test technique :-Equivalence partitioning (valid/invalid input)

Boundary value analysis :-(254,255,256 characters )

Error guessing :- script injection , empty input

Steps :-

1. Navigate to custom field creation module.

2. Click "Create New Field".

3. Select "Text (Single Line)" as the field type.

4. Enter field label (e.g., “User Name”).

5. Save the field.

6. Use the field in a form and input valid and invalid data.

Pre condition:-

1.User must be logged in with admin or field-config creation permissions.

2. Navigation to the "Custom Fields" or "Field Configuration" page is available.

Test data :-

1.Field Label: User Name

2.Field Type: Text (Single Line)

3.Input Text: JohnDoe123

4.Max Length: 255 characters

5.Invalid Input: Empty string, >255 characters

Severity :Medium

Priority :high

Expected result :-

1.Field should be created successfully with label .

2.The field should accept alphanumeric input (e.g., "JohnDoe123").

3. Input should be restricted to 255 characters.

Actual Result :-

Pass :- 1.Field should be created successfully.

2.The field should accept alphanumeric input (e.g., "JohnDoe123").

3. Input should be restricted to 255 characters.

fail :-

1.Field should not accept more than 255 characters.

2.Field should display validation if input is empty (if required).

------------------------------------------------------------------------------------------------------------------------------------

72.Validate that a custom field of type "Number" only accepts numeric input and rejects non-numeric values.

Bug Description:- Field allowed letters or did not show validation for incorrect input.

PIM\_config\_CF\_Number

Test type : Functional testing

Test method :- Manual Testing

Test techniques :-

Equivalence partitioning (valid v/s invalid )

Boundary value analysis (max,min values)

Error guessing (entering symbols, letters ,script injection)

Steps :-

1.Log in as admin/config user Access granted

2 Navigate to Custom Fields > Create New Field Field creation screen opens

3 Select field type: Number, enter label (e.g., “Order Quantity”) Field type set; label accepted

4 Save the field Field created successfully

5 Add the field to a form layout (e.g., Ticket or Order form) Field appears in form builder

6 Open the form in user view Field displayed as numeric input

7 Enter valid numeric values like 123, -45, 0.5 Input is accepted

8 Try entering abc, 12ab, @#! System shows validation error

9 Submit the form with valid input Form submits successfully

Pre condition:-

1.User is logged in with admin or custom field creation permissions.

2.Access to the Custom Field Configuration module is available.

3.At least one form or module (e.g., Ticket, Order, Profile) exists where custom fields can be added.

4.The platform supports the Number field type (integer or decimal).

Test data :-

123\_ Valid Number\_ Accepted

-45.67 \_Valid Decimal\_ Accepted (if decimals allowed)

abc\_ Alphabetic\_ Rejected

12ab\_ Alphanumeric\_ Rejected

!@# \_Special Characters\_ Rejected

(blank)\_ Empty\_ Error if required

Severity :medium

Priority :high

Expected result :- Field accepts valid numeric input; rejects letters, symbols, or mixed characters."

Actual result :- Pass:- Field accepted only numbers; rejected alphabets and special characters.

Fail:- Field allowed letters or did not show validation for incorrect input.

------------------------------------------------------------------------------------------------------------------------------------

73.Validation error message is displayed when the custom label fields includes special character .

Bug Description:- Field saved with special characters (no validation)

PIM\_config\_CF\_custom

Test type :- Negative testing  
Test method :- manual  
Test techniques :- 1.Error testing  
(common user mistakes like special characters )  
2. input validation testing  
3.Boundary value analysis

Steps:-

1.Log in to the application with admin access.  
2.Navigate to the Custom Fields section.  
3.Click on Create New Field.  
4.Select field type (e.g., Text).  
[5.In](http://5.In) the Field Label, enter: @Name!  
6.Fill in any other required field values.  
7.Click Save.

Pre condition :- 1.The user is logged in with administrator or custom field management permissions.  
2.The Custom Field Creation page is accessible from the admin/configuration panel.  
3.The platform supports manual entry for field labels.  
4.System has defined validation rules for field labels (e.g., allows only letters, numbers, and limited symbols).

Test Data :- Field \_Input  
Field Type \_Text (Single Line)  
Field Label \_@Name!

Severity :- Medium- May not break the system but can affect UI consistency or data integrity.

Priority :- Medium to High- Should be restricted or sanitized for cleaner UI and safe database entries.

Expected Result :- The system should either:  
1.Show a validation error (e.g., “Special characters are not allowed”),  
Or sanitize the label (e.g., auto-remove special characters),  
Or block submission until corrected.

Actual Result :-

Pass:Validation error shown; field not created

Fail:Field saved with special characters (no validation)

--------------------------------------------------------------------------------------------------------------

1. Validate error message is displayed when custom field is created with duplicate label.

Bug Description:- Duplicate label allowed, field created

PIM\_config\_CF\_custom

Test type :- Negative testing  
Test method :- Manual testing  
Test techniques :-  
1.Error guessing :anticipating user mistakes .  
2. Input validation testing  
3. State transition testing

Steps:-

1.Log in as a user with admin or config permissions.  
2.Navigate to the Custom Fields section.  
3.Click Create New Field.  
4.Select any field type (e.g., Text).  
[5.In](http://5.In) the Field Label, enter: Customer ID.  
6.Click Save.

Pre condition:-

1.The user is logged in with admin/config access.  
2.The Custom Fields module is accessible.  
3.A custom field with the label “Customer ID” already exists in the system.  
4.The platform enforces unique field labels for each form/module.

Test data :- Field\_ Value  
1.Field Type \_Text (Single Line)  
2.Field Label \_Customer ID

Severity :- Medium: Doesn’t crash the system but affects configuration integrity.

Priority :- High: Important to prevent confusion or conflict in form submissions/data.

Expected Result :- 1.The system checks for duplicate field labels.  
2.Field creation is blocked.  
3.A validation error is shown: “Field name already exists” or similar.

Actual Result :- Pass:Duplicate label blocked, error shown  
Fail:Duplicate label allowed, field created

------------------------------------------------------------------------------------------------------------------------------------

1. Validate that the system prevents SQL injection by sanitizing or rejecting malicious input in dropdown option values during custom field creation.

Bug Description:- Input accepted and saved without validation

(create manually , delete through SQL query )

test type ; negative testing  
Test method : manual Testing  
Test technique :Injection testing , Error guessing

Steps:-

1.Log in as a user with Admin privileges.  
2.Go to Custom Fields or Field Configuration.  
3.Click Create New Field.  
4.Select field type: Dropdown (Single Select).  
5.Enter Field Label: Status.  
6.Enter the following Dropdown Option:  
1'); DELETE FROM fields; --  
7.Click Save.

Pre condition:-

1.User is logged in with admin/config permissions.  
2.The Custom Field Creation module is accessible.  
3.The system supports manual entry of field labels and dropdown options.

Test data :-

Field\_ Input  
1.Field Type \_Dropdown (Single Select)  
2.Field Label DROP TABLE users;  
3.Dropdown Options DELETE FROM fields;

Severity :- critical

Priority :- high

Expected Result :- 1.The system blocks or escapes the SQL-like input.  
2.Shows a validation error: “Invalid characters detected in dropdown options”.  
3.Field is not saved with unsafe input.  
[4.No](http://4.No) database actions are performed.  
5.If logging is enabled, the input attempt may be flagged as a security warning.

Actual Result :- Pass:System blocks the input; no field created

Fail:Input accepted and saved without validation

----------------------------------------------------------------------------------------------------------------

1. Validates error message is displayed when system rejects non-numeric when using custom field of type "Number"

Bug Description:- The form allows submission with invalid (non-numeric) data.

PIM\_Config\_CF

Test method : Manual testing  
Test Techniques :- Input validation, Boundary value Testing , Error guessing

Steps:-

1.Navigate to a form that uses the Number field (e.g., Ticket or Profile form).  
2.Locate the number field (e.g., "Age", "ID Number").  
3.Enter a non-numeric value such as abc into the number field.  
4.Submit or try to move focus away from the field.

Pre condition:-

1.User is logged in with permission to create and use forms with custom fields.  
2.A Number-type custom field is already created and visible on a test form (e.g., "Create Ticket" or "User Profile").

Test data :- Field Type\_ Input  
1.Number \_abc (invalid)

Severity :medium

Priority :- High

Expected Result :- System displays a validation error, such as: “Please enter a valid number.”

Input is not accepted, and the form cannot be submitted until corrected.

Actual Result :- Pass:1.The system detects and blocks non-numeric input (abc) in the Number field.  
2.A validation error message appears, such as:  
“Please enter a valid number.”  
The form does not allow submission until the input is corrected.  
The invalid value is not saved in the database.

Fail:- 1.The system accepts the non-numeric input (e.g., abc) in the Number field without showing an error.  
2.The form allows submission with invalid (non-numeric) data.  
3.The value is saved incorrectly or causes an error in the backend/database.

---------------------------------------------------------------------------------------------------------------------------------------------------

77.Validate no authorization access is granted and user not created with escalated privileges via SQL injection in Email field

Bug Description:- If user is created with elevated privileges or login is bypassed

PIM\_Configouration\_Dataimport

Test type :- negative

Test technique :- Black box testing

Test method :- Security

Steps:-

1.Create a .csv or .xlsx import file

2.In the Email field, enter admin' --

3.Fill remaining fields validly (e.g., Employee ID, Name)

4.Import the file

5.Check if user is created as admin, or if login bypass occurs using that email

6.Check logs and database for unauthorized user creation or role escalation

Pre condition:-

1.Authentication module is active (login or import flow includes email/user role fields)

2.Import or login accepts email input from user or file

Test data :-

Example CSV:

First Name, Last Name, Email, Employee ID

Test, User, admin' --, E007

Severity :- Critical – Can lead to unauthorized access or privilege escalation

Priority :- High – Needs urgent fix if found vulnerable

Expected Result :-

1.No unauthorized access is granted

2.User is not created with escalated privileges (e.g., admin role)

Actual result :-

Pass: – If input is rejected, sanitized, or safely stored without affecting access controls

Fail:-If user is created with elevated privileges or login is bypassed

------------------------------------------------------------------------------------------------------------------------------------

78.Validate Selected reporting method is saved and reflected in the employee’s profile or reporting structure.

Bug Description:- If not saved, error appears, or value is lost

PIM > Employee Information / Reporting Methods

Test Technique :- Black Box Testing

Test method :- Manual / UI-based Functional Testing

Steps:-

1.Login as Admin

2.Navigate to PIM > Employee List

3.Edit an employee’s details

4.Go to the Job or Reporting Structure section (as per UI)

5.Assign the reporting method: Matrix Reporting

6.Click Save

Pre condition:-

1.User is logged in as Admin/HR

2.At least one employee exists

3.At least one reporting method is configured

Test data :-

1.Employee: John Doe (Emp ID: E123)

2. Reporting Method: Matrix Reporting

Severity :-

Medium – Impacts reporting hierarchy and structure

Priority :-

High – Required for correct org structure setup

Expected Result :-

Selected reporting method is saved and reflected in the employee’s profile or reporting structure

Actual result :-

Pass – If reporting method is correctly assigned and saved

Fail – If not saved, error appears, or value is lost

------------------------------------------------------------------------------------------------------------------------------------

1. Validate Error message is displayed on submiting Timesheet with missing hours

Bug Description:- [No](http://2.No) warning is shown, and the action silently fails.

Test Type :-Functional / Negative  
Test Method :-Manual (UI-based)  
Test Technique:- Black-box

Steps:-1.Log in as the employee.  
2.Navigate to Time > My Timesheets.  
3.Click Edit for the current or target week (e.g., July 8–14, 2025).  
4.Enter valid working hours on some days (e.g., Mon: 8, Tue: 0, Wed: blank, rest: blank).  
5.Click Submit.  
Observe whether:  
6.System allows submission,  
Or a warning appears about missing daily hours.

Pre condition:-1.Employee is logged in and has access to at least one valid project and task.  
2.A new timesheet is available for the current or selected week.  
3.The timesheet is in editable state (Draft or not yet submitted).

Test data :-Field\_ Value  
1.Employee Name Alex Kim  
2.Employee ID EMP034  
3.Week Range\_ July 8–14, 2025  
4.Project \_Customer Onboarding  
5.Task \_Client Setup  
6.Entered Hours \_Mon: 8, Tue: blank, Wed–Sun: blank  
7.Expected Behavior \_Warning or conditional submission  
8.Status Expected \_Blocked or Submitted with partial data

Severity :- medium

Priority :-medium

Expected Result :-System behavior should follow defined business rules:  
1.If partial submission is allowed: Timesheet is submitted with missing hours, and status changes to "Submitted".  
2.If full entry is mandatory: User receives a validation warning, e.g., “Please complete all days before submitting.”  
3.UI should not crash or show unexpected behavior.

Actual Result:-Pass:-  
1.System behaves consistently with business logic:  
2.Either permits partial entry or correctly blocks with an informative error message.  
3.UI shows clear feedback (no silent failures).  
4.Entered data is saved or preserved on error (no data loss).

Fail:-  
1.Timesheet submits with missing data when it shouldn't.  
[2.No](http://2.No) warning is shown, and the action silently fails.  
3.Error message is unclear or UI crashes.  
4.Blank fields are incorrectly treated as valid zero hours.

---------------------------------------------------------------------------------------------------------------------------------------------------

1. Validate Submit Timesheet with Missing Hours (Partial Submission Allowed)

Bug Description:- System blocks submission unexpectedly.

Test Type Functional / Negative-to-Positive  
Test Method Manual (UI)  
Test Technique Black-box

Steps:-1.Log in as the employee.  
2.Navigate to Time > My Timesheets.  
3.Click Edit for a week (e.g., July 8–14, 2025).  
4.Enter hours for only 2–3 days (e.g., Mon: 8, Tue: 0, rest: blank).  
5.Click Submit.  
6.Observe system behavior.

Pre condition:-1.Employee is logged in with access to at least one project/task.  
2.Timesheet for selected week is editable and not yet submitted.  
3.Blank days are acceptable based on organizational policy.

Test data :-Field Value  
1.Employee Alex Kim (EMP034)  
2.Week July 8–14, 2025  
3.Project Client Integration  
4.Task Technical Support Setup  
5.Entered Hours Mon: 8, Tue: blank, Wed: 6, Thu–Sun: blank  
6.Expected Status Submitted  
7.Warning Expected Optional (non-blocking)

-- Check status of timesheet  
SELECT \* FROM ohrm\_timesheet  
WHERE employee\_id = (SELECT emp\_number FROM hs\_hr\_employee WHERE employee\_id = 'EMP034')  
AND start\_date = '2025-07-08';

-- Check stored items (note blank days may show as 0 or be absent)  
SELECT \* FROM ohrm\_timesheet\_item  
WHERE timesheet\_id = [FROM\_ABOVE\_QUERY];

Severity :-medium

Priority :-medium

Expected Result :-1.Submission is allowed even with blank days.  
System either:  
2.Accepts submission silently (treats blanks as 0 hours), or  
Displays a non-blocking warning, e.g.,  
“Some days are blank. Do you want to continue?”  
3.Timesheet status changes to Submitted.  
4.Data is saved correctly with blank or 0-hour entries in the backend.

Actual Result:-Pass:  
1.Timesheet is submitted without blocking errors.  
2.Blank fields are stored as 0 or null depending on backend design.  
3.Status shown as Submitted.  
4.UI does not crash or show inconsistent state.

Fail:-  
1.System blocks submission unexpectedly.  
2.Data saved incorrectly (e.g., blanks converted to invalid characters).  
3.Status remains Draft or errors out.  
4.Confirmation message is missing or misleading.

------------------------------------------------------------------------------------------------------------------------------------

81.Validate Handle partial week leave with working days in timesheet

Bug Description:- Leave entries are not populated despite approved leave.

Time > My Timesheets + Leave Module

Test Type: Integration, Functional

Test Method :Manual

Test Technique: Black-box

Severity :- High

Priority :- High

Pre condition:-

1.Employee has approved leave for part of the week (e.g., Monday–Wednesday).

2.Remaining days (e.g., Thursday–Friday) are working days.

3.Leave and time modules are integrated.

4.Timesheet is editable.

Test Data:-

Day Type Task Hours

1.Monday\_ Leave\_ Leave – Annual \_8

2.Tuesday\_ Leave\_ Leave – Annual \_8

Wednesday \_Leave \_Leave – Annual\_ 8

Thursday\_ Work \_Project – ABC\_ 7.5

Friday\_ Work\_ Project – XYZ\_ 8

Total — — 39.5

-- Check timesheet entries

SELECT day, project\_name, duration

FROM ohrm\_timesheet\_item

WHERE timesheet\_id = [TIMESHEET\_ID];

-- Validate leave overlap for partial days

SELECT \* FROM ohrm\_leave

WHERE employee\_id = [EMP\_ID]

AND leave\_status = 2

AND from\_date >= '2025-07-07' AND to\_date <= '2025-07-11';

---------------------------------------------------------------------------------------------------------------------------------------------------

82.Validate on occurance of error message Submit timesheet with missing hours for some days

Bug Description:- Incorrect total hours (e.g., blank fields not counted as 0).

Time > My Timesheets

1.Test Type Negative / Validation Test  
2.Test Technique Black-box

Steps:-1.Navigate to Time > My Timesheets.  
2.Select the current week and click Edit.  
3.Enter valid hours for some days (e.g., Monday and Wednesday).  
4.Leave other days blank (e.g., Tuesday, Thursday, Friday).  
5.Click Submit.

Pre condition:-1.Timesheet for current week is in Draft status.  
2.User has access to at least one active project/task.  
3.Employee is logged in and on the Edit Timesheet screen.

Test data :-Day\_ Task \_Hours  
1.Monday \_Project A – Dev 8  
2.Tuesday (left blank)  
3.Wednesday Project A – Dev 6  
4.Thursday (left blank)  
5.Friday \_Project A – Dev 7

Optional:-  
SELECT \* FROM ohrm\_timesheet\_item  
WHERE timesheet\_id = [ID]  
ORDER BY date;

1.Ensure rows exist for all days (Mon–Fri).  
2.Days with no input should show duration = 0.

Severity :-medium

Priority :-medium

Expected Result :-Option A:  
1.Warning / Confirmation Prompt  
2.System displays a warning or confirmation, such as:  
3.“Some days have no hours entered. Do you still want to submit?”  
4.If confirmed, timesheet is submitted and blank days are treated as 0 hours.

Option B:  
1.Validation Error  
2.System blocks submission.  
Error message appears:  
1.“All days must have hours entered.”

Actual Result:-Pass :  
1.System recognizes the missing days.  
2.Appropriate behavior occurs (either warning or error, as configured).  
3.If submitted, the timesheet status = Submitted, and missing days have 0 hours in the backend.

Fail:-  
1.Submission is accepted without warning but missing data is lost or corrupt.  
2.Incorrect total hours (e.g., blank fields not counted as 0).  
3.System crashes or behaves unpredictably on blank inputs.

------------------------------------------------------------------------------------------------------------------------------------

1. Validate by Editing an employee’s timesheet before approval

Bug Description:- Changes are not reflected after saving

Time > EmployeeTimesheet

1.Test Technique Black Box Testing

2.Test Method Manual Testing

3.Test type : Negative testing

Pre conditions:-

Employee has a timesheet in Draft or Submitted status

1.User (employee or manager) has edit permissions

2.Timesheet is not yet Approved

Test Data :-

Employee Name / ID: Mike Lee / EMP789

1.Timesheet Week: 15-Jul-2025 to 21-Jul-2025

2.Original Entry: 8 hours on Task “Development”

3.Edited Entry: Change to 7.5 hours on Task “Testing”

Severity :- Major – Editing errors can affect payroll, but system remains stable

Priority :- Medium – Important but less critical than approval/rejection

Expected Result :-

Timesheet updates are saved successfully

1.Changes are visible immediately in the UI

2.Status remains "Draft" or returns to "Submitted" depending on workflow

3.No system error or data loss occurs

Actual result :-

Pass:-

1. Edited time entries are saved correctly

2.No UI or backend errors

3.Modified data is accurate and visible

4.Status changes accordingly if workflow requires resubmission

Fail :-

1.Edit/save fails or throws an error

2.Changes are not reflected after saving

3.System allows editing of an approved timesheet (should be restricted)

4.Partial or corrupt data is saved

-----------------------------------------------------------------------------------------------------------------------------------

84.Validate Attempt to view or create timesheet for a future date range

Bug Description:- System allows adding/editing future entries

Test technique :- Black Box Testing

Test method :- Manual testing

Test type : Negative Testing

Steps :-

1. Go to Employee Timesheet

2. Select a future week (e.g., "05-Aug-2025 to 11-Aug-2025")

3. Attempt to view or create

Pre condition:-

System should restrict timesheet entry for future weeks

Test Data :-

Date Range: 05-Aug-2025 to 11-Aug-2025

Severity :- Minor

Priority : low

Expected Result :-

System shows message: “Future timesheets cannot be accessed” or disables future weeks

Actual result :-

Pass:

1.System prevents access to future timesheets gracefully

Fail:

1.System allows adding/editing future entries

2.No validation shown

----------------------------------------------------------------------------------------------------------------------------------------------

1. Validation message like “Hours cannot be negative” on Entering negative hours in a timesheet field

Bug Description:- System accepts the negative value

Test type : Negative Testing  
Test Technique :Black Box Testing  
Test Method: Manual Testing

Steps:-Navigate to Time > My Timesheet

Select the week 01-Jul-2025 to 07-Jul-2025

Enter -5 in the hour field for any day (e.g., Wednesday)

Attempt to save or submit the timesheet

Post condition:-   
1.Negative value is not stored  
2.Field remains editable for correction

Pre condition:-1.User is logged in (Employee role)  
2.Timesheet is in Draft status  
3.User is on the timesheet entry screen

Test data :-1.Employee: Priya Mehta / EMP210  
2.Week: 01-Jul-2025 to 07-Jul-2025  
3.Entry: -5 hours on 03-Jul-2025 (Wednesday)

Severity :-Major – Can lead to incorrect payroll calculations and invalid data

Priority :-Medium

Expected Result :-1.System shows an error or validation message like “Hours cannot be negative”  
2.Entry is rejected and not saved

Actual Result:-Pass:  
1.Validation is triggered immediately or on submission  
2.User is prevented from saving negative hours

Fail:-  
1.System accepts the negative value  
2.Timesheet saves/submits without error  
3.System crashes or behaves unpredictably

------------------------------------------------------------------------------------------------------------------------------------

85.Validate System should redirect to the login page on Attempting to access the timesheet module directly via URL without authentication

Bug Description:- [No](http://2.no/" \o "http://2.No) redirect or invalid session handling

Test type : Negative Testing

Test Technique: Black Box Testing

Test Method: Manual Testing

1.Open browser in incognito/private mode

2. Navigate directly to the timesheet URL (e.g., /time/viewMyTimesheet)

3. Observe system behavior

Note : 1.User must log in to access timesheet

1. No data exposure or session misuse occurs

Pre condition:-

User is not logged in to OrangeHRM

1.Browser is open and cookies/session are cleared or expired

Test Data :-

URL: <https://<your-orangehrm-url>/index.php/time/viewMyTimesheet>

Severity :-

Critical – Unauthorized access could expose sensitive employee data

Priority :- High

Expected Result :-

System should redirect to the login page

1. Timesheet content should not be visible without authentication

Actual Result :-

Pass:

1.Redirect occurs immediately

2.Timesheet module is not accessible without logging in

fail:

Page loads without authentication

1.Any employee data is visible or editable

2.No redirect or invalid session handling

------------------------------------------------------------------------------------------------------------------------------------

86.Validate System should reject invalid input and display an appropriate error message like: "Invalid time format.

Bug description:- System accepts the invalid time format OR crashes/hangs without validation

1.Test Type :Negative Testing, Validation Testing, UI Testing

2.Test Technique : Boundary Value Analysis, Equivalence Partitioning, Error Guessing

Test Steps :-

1.Log in to the timesheet application

2.Go to the time entry section

3.Attempt to enter "25:61" as the time duration

4.Observe the system response

5.Repeat the process with "abc" and other invalid formats

6.Try to submit the entry

7.Verify if the system blocks submission and shows an error |

Pre condition :-

User is logged in

1.Timesheet is open

2.Time entry field is visible and editable

Test Data :-

Invalid Input: "25:61", "abc", "99:99", "##:##"

Severity :- High

Priority :- High

Expected Result :-

System should reject invalid input and display an appropriate error message like: "Invalid time format. Please use HH:MM."

Actual result :-

Pass:

System displays a clear error message and does not allow the user to submit the invalid entry

Fail:System accepts the invalid time format OR crashes/hangs without validation

------------------------------------------------------------------------------------------------------------------------------------

87.Validate Cross-Midnight Shifts with Shift-End Auto Punch-Out

Time\_Attendance

1.Test Type :Functional, Time Boundary, Auto Processing

2.Test Technique: Boundary Value Analysis

Test Steps :-

1.Log in as Admin.

2.Configure a night shift from 10:00 PM to 6:00 AM (crossing midnight).

3.Enable the “Auto Punch-Out at Shift End” setting.

4.Save the configuration.

5.Log in as the Employee.

6.Punch In at 10:00 PM on Day 1.

7.Do not perform manual Punch Out.

8.Wait until next day morning (6:00 AM, Day 2).

9.Log in as Admin and review the employee's attendance record.

Pre condition :-

Shift defined: 10:00 PM to 6:00 AM (next day)

1. "Auto Punch-Out" and "Shift End" are enabled

2.Employee exists and is assigned to the shift

Test Data :-

Punch-In Time: 10:00 PM

1. Expected Auto Punch-Out Time: 6:00 AM (next day)

Severity :- High

Priority :- High

Expected Result :-

1.The system should automatically generate a Punch-Out at 6:00 AM on Day 2 (not Day 1).

2.The entry should reflect the correct date rollover and comply with the shift's end boundary.

Actual result :-

Pass:

1. Punch-out is logged on the correct next calendar day.

2. System clearly shows the shift spanning two days with accurate date stamps

Fail:-

1.Auto punch-out is recorded on the same day as punch-in (Day 1).

2.Incorrect or missing punch-out time.

3.Inaccurate shift duration or calculation errors in reports.

------------------------------------------------------------------------------------------------------------------------------------

88.Validate Simultaneous Admin Edits on Attendance Configuration

(Configouration conflict )

Bug Description:- No warning or conflict detection is triggered.

1.Test Type :Functional, Concurrency, Race Condition Handling

2.Test Technique :Error Guessing, Concurrency Testing

Steps :-

1.Log in as Admin A and navigate to Time → Attendance → Configure.

2.Log in as Admin B in a separate session or browser; open the same configuration page.

3.Admin A: Enable Auto Punch-Out, then wait.

4.Admin B: Disable Auto Punch-Out, then click Save.

5.Admin A: Now click Save without refreshing the page.

6.Check the final value of Auto Punch-Out setting in the system.

Pre cndition :-

Two active admin accounts

1. Both have permission to edit attendance settings

2.“Auto Punch-Out” setting is available

Test Data :-

1.Admin A: enables “Auto Punch-Out”

2.Admin B: disables it (conflicting action)

Severity :- High (risk of misconfiguration or silent data corruption)

Priority :- medium-high

Expected Result :-

1.System should either:

a. Show a warning or error that another admin has updated the config.

b. Auto-reload the config page to fetch the latest saved values.

2. Only one config version should persist.

Actual result :-

Pass:

1.System detects and prevents silent overwrite.

2.Config reflects latest saved value clearly.

3. Conflict is handled gracefully with UI prompt or server-side lock.

Fail:

1.Both admins can overwrite each other’s changes silently.

2. Final config is unpredictable.

3. No warning or conflict detection is triggered.

------------------------------------------------------------------------------------------------------------------------------------

1. Validate Disabled Attendance Module While Config Rules Remain Active

Bug Description:- System executes attendance rules despite module being OFF.

1.Test Type :Functional, Negative Testing, System Control

2.Test Technique :Error Guessing, State Transition Testing

Test Steps :-

1.Log in as Admin.

2.Navigate to: Admin → Configuration → Modules.

3.Disable the Attendance module using the global toggle.

4.Save the configuration.

5.Log in as Employee.

Attempt to access the following:

a. Time → Attendance

b. Punch In/Punch Out screen

c. View/edit previous attendance

6.Monitor for error messages or blocked access.

Pre condition :-

Attendance module is active

1. Config rules (e.g., IP restrictions, auto punch-out) are enabled

Test Data :-

Attendance module toggle = OFF

1.IP restriction = ON

2.Shift = Active

Severity : High

Priority : High

Expected Result :-

1.All attendance features should be completely inaccessible to users.

2.Employees should see a clear message like “Attendance module is currently disabled.”

3.System should not execute background actions (e.g., auto punch-out) while module is OFF.

Actual result :-

Pass:

1.Attendance screens are blocked.

2. No time-related actions are processed.

3. Active rules like IP restrictions or shift end logic are suspended or ignored.

Fail:

1.Attendance screens are blocked.

2.No time-related actions are processed.

3. Active rules like IP restrictions or shift end logic are suspended or ignored.

------------------------------------------------------------------------------------------------------------------------------------

90.Validate Dynamic Attendance Config per Pay Grade (e.g., IP Restriction Exceptions)

Bug description:- IP restriction applied incorrectly (e.g., blocks managers).

Managers can punch in/out from any IP, Staff can punch only from allowed IPs

1.Test Type :Functional, Role-Based Access Control, Integration

2.Test Technique: Equivalence Partitioning, Decision Table Testing

Steps :-

1.Admin logs in → Configure attendance policy:- Enable IP restriction for staff (allowed IPs set) - Exempt managers from IP restriction. Save settings.

2.Employee A (Manager) logs in from an unapproved IP (e.g., home or mobile hotspot).

3.Employee A attempts to punch in/out → Verify if system allows punch.

4.Employee B (Staff) logs in from the same unapproved IP.

5.Employee B attempts to punch in/out → Verify if system blocks punch with error message.

6.Repeat punch attempts from approved IP for both employees → Verify allowed access.

Pre condition :-

1.Multiple pay grades defined in the system (e.g., Manager, Staff).

2.Attendance config supports policy linking to pay grades.

3.IP restriction enabled for general users.

4.Managers are excluded from IP restriction (allowed from any IP).

Test Data :-

1.Employee A: Pay Grade = Manager

2.Employee B: Pay Grade = Staff

3.IP restriction enabled for non-managers

4.Allowed IP list: Company office IP only

Severity :- High

Priority :- High

Expected Result :-

1.Managers can punch in/out from any IP, regardless of restriction.

2.Staff can punch only from allowed IPs; others blocked.

3.System enforces dynamic config based on pay grade linked policies.

4.Proper messages displayed on block or success.

Actual result :-

Pass:

1.Policy enforcement aligns perfectly with pay grade assignment.

2.No leakage of permissions across grades.

3.Audit logs show policy applied per user role.

Fail:

1. IP restriction applied incorrectly (e.g., blocks managers).

2. Staff allowed to punch from unapproved IP.

3. System ignores pay grade linkage for attendance config.

------------------------------------------------------------------------------------------------------------------------------------

91.Validate Real-Time Enforcement of Config Change While Editing Past Timesheet

Bug Description:- Employee is still able to save changes after config is disabled.

1.Test Type: Functional, State Consistency, Permission Enforcement

2.Test Technique :State Transition Testing, Negative Testing

Steps :-

Step 1 (Setup):

1.Admin ensures the editable time config is ON.

2.Employee logs in → navigates to Time > MyTimesheet.

3.Employee selects last week’s timesheet (e.g., July 1–7) → starts editing time entries.

Step 2 (Config Change During Edit):

1.While employee still has the edit view open:

2.Admin toggles Allow time editing for past periods = OFF.

3.Clicks Save.

Step 3 (Attempt Save):

1. Employee finishes editing and clicks Save/Submit.

Pre condition :-

1.An employee has submitted or is editing their past timesheet (e.g., last week: July 1–7, 2025).

2.Config setting Allow employee to edit past timesheets is initially ON.

3.Employee is actively editing the timesheet in the browser.

4.Admin has access to Attendance Configuration panel.

Test Data :-

Field \_Value

1.Employee Username \_john.smith

2.Week to Edit\_ July 1–7, 2025

3.Time Entry \_9:00 AM – 6:00 PM

4.Config Before \_Editable Time = ON

5.Config After \_Editable Time = OFF

6.Action Attempted\_ Save timesheet edits

7.Expected Outcome \_Save blocked with message

Severity :-

High – Can lead to unauthorized edits in sensitive time data

Priority :-

High – Needs validation in every release that involves time config

Expected Result :-

Aspect \_Behavior

1.UI Response \_On Save: User gets error message → “Editing past entries is no longer allowed.”

2.Real-Time Effect\_ Fields may auto-lock or UI reloads with current config state

3.Save Behavior\_ Save is denied if config changed mid-session

4.Audit Log \_System logs attempted edit with message: "Blocked by config rule"

Actual result :-

Pass:

1.System correctly detects config change in real-time or on save attempt.

2.User is not allowed to save changes once edit permission is revoked.

3.Clear, user-friendly message is shown.

4.Audit trail captures blocked attempt (if applicable).

Fail:A

1.Employee is still able to save changes after config is disabled.

2.No error or validation message shown.

3.Audit/logs do not reflect unauthorized save attempt.

92.Validate UI When Attendance Module Is Disabled

Bug Description:- Employee is able to save or submit data.

1.Test Type: Functional, UI/UX Consistency, Permission Validation

2.Test Technique :Negative Testing, State Transition Testing

Steps :-

1.Log in as Admin.

2.Go to System > Configuration > Modules (or similar).

3.Locate the Attendance/Timesheet module toggle.

4.Set Attendance module = OFF.

5.Save and apply the changes.

6.Log in as Employee (or refresh an existing employee session).

7.Navigate to Time > MyTimesheet.

Pre condition :-

1.Attendance module is initially enabled and working.

2.Employee has active access to Time > MyTimesheet.

3.Admin has access to global module configuration (System → Configuration/Modules).

Test data :-

Field\_ Value

1.Attendance Module Status \_Disabled

2.Employee Name\_ sarah.thomas

3.Timesheet Period\_ July 1 – July 7, 2025

4.UI Behavior Expected Fields\_ disabled, warning shown

5.Backend Behavior Expected\_ Save blocked / 403 error

Severity :- Medium – Confusing UI can lead to incorrect user actions

Priority :- High – Misleading UI undermines reliability and user trust

Expected Result :-

UI Element \_Expected Behavior

1.Time entry fields\_ Grayed out / disabled

2.Action buttons (Save, Submit)\_ Hidden or disabled

3.Warning message \_Clear notification shown: “Attendance module is currently disabled.”

4.Navigation\_ Optional: Redirect to dashboard or prevent access to MyTimesheet

5.API/API response (optional) \_403 or similar “module disabled” response if accessed via API

Actual Result :-

Pass:

1.Employee cannot edit or submit any timesheet data.

2.System UI reflects disabled state clearly and consistently.

3.No backend actions (save, update) succeed for disabled module.

4.Warning message is user-friendly and instructive.

Fail :

1.Time entry fields remain editable.

2.Employee is able to save or submit data.

3.No warning or feedback is provided.

4.Behavior differs across web and mobile app.

------------------------------------------------------------------------------------------------------------------------------------

93.Validate Mobile vs Desktop Behavior Divergence

Punch-in works on desktop. On mobile app (or PWA), the time field behaves inconsistently, even though both platforms use the same configuration.

* Bug Description:- Punch-in fails only on mobile or only on desktop.

Time > Punch In (Web + Mobile App / PWA)

1.Test Type: UI Consistency, Functional

2.Test Technique: Comparison Testing, Equivalence Partitioning

Steps :-

1. Log in as Employee C via desktop browser.

2. Navigate to Time > Punch In.

3. Verify time field (e.g., defaults to current time, editable, accepts input).

4. Log out and switch to mobile app/PWA.

5. Navigate to the same Punch In screen.

6. Observe time field behavior (editable, default values, time format, validation).

7. Try submitting punch-in on both platforms.

8. Compare behavior, validations, and UI handling.

Pre condition :-

Attendance punch-in is enabled.

1.Same shift config applies to all platforms.

2.User has access to both desktop browser and mobile app/PWA.

3.No custom mobile overrides are applied.

Test Data :-

Field Value / Description

1.Employee ID EMP102

2.Username emp102@example.com

3.Shift Timing 09:00 AM – 06:00 PM

4.Time Field Config - Editable: ✅ Yes

- Default: ✅ Current system time

- Format: ✅ 24-hour

5.Punch Location Any (Location not restricted in this test)

Device 1 (Desktop) Google Chrome v125 on Windows 11

Device 2 (Mobile) - Android 13 with Company Mobile App

6. PWA via Chrome on iOS Safari

7.Expected Time Value 09:00 (if opening screen at 9 AM)

8.Should match across both platforms

9.Network Same stable Wi-Fi for both devices (to rule out sync/timeout issues)

10.Time Field Format 24-hour on both devices

11.Timezone UTC+05:30 (IST)

12.App Version Mobile App v3.2.1

13.PWA build version should match desktop web

Severity :- High – Inconsistent punch behavior causes user confusion and errors.

Priority :- High

Expected Result :- 1.Time field behavior is consistent across desktop and mobile:

- Default values

- Editable vs non-editable

- Validation behavior

- Time format (12h/24h)

1. Punch-in works as expected on both.

Actual result :- 1.Time field behavior is consistent across desktop and mobile:

- Default values

- Editable vs non-editable

- Validation behavior

- Time format (12h/24h)

2. Punch-in works as expected on both.

------------------------------------------------------------------------------------------------------------------------------------

1. Validate Mobile vs Desktop Punch-in UI Consistency

Bug Description:- .Time field is editable on one platform but not the other.

Time > Punch In (Desktop Web App vs Mobile App / PWA)

Punch-in works correctly on the desktop web app, but on mobile app (or PWA), the time field behaves inconsistently (e.g., not editable, different default, wrong format), even though the same configuration is applied.

Test Type :UI Consistency, Functional

Test Technique :Comparison Testing, Equivalence Partitioning

Test steps:-

1.Log in as Employee on a desktop browser.

2.Navigate to Time > Punch In.

Note:

3.Is the time field editable?

4.Does it default to current time?

5.What format is shown (12h or 24h)?

6.Submit a punch-in entry and confirm it works.

7.Log in as the same user on the mobile app or PWA.

8.Go to the same Punch In screen.

Observe:

9.Time field behavior: editable or locked?

10.Default value?

11.Time format?

12.Submit punch-in entry and compare behavior with desktop.

Pre condition:-

1.Punch-in is enabled in the system.

2.Employee has access to both desktop and mobile platforms.

3.Time field configuration is centrally managed (no device-specific overrides).

4.Same shift timing applied to user profile.

Test Data :-

Field Test Data

1.Employee ID EMP102

2.User Login Credentials Username: emp102@example.com

3.Password: \*\*\*\*\*\*\*\*

4.Shift Timing 09:00 AM – 06:00 PM

5.Time Field Configuration - Editable: ✅ Yes

- Default Value: ✅ Current System Time

- Format: ✅ 24-hour

6.Punch Location Setting Not restricted (geo-fencing disabled for this test)

7.Timezone UTC+05:30 (IST)

8.Device 1 – Desktop - Browser: Chrome v125

- OS: Windows 11 / macOS Sonoma

- Resolution: 1920x1080

9.Device 2 – Mobile - Android 13 (App v3.2.1) OR iOS 17 (PWA via Safari)

- Resolution: ~1080x2400

10.Network Same Wi-Fi network or stable mobile data (to remove latency/internet variables)

11.Test Time Window Between 8:55 AM and 9:10 AM (to verify current time default is accurate and matches)

12.Expected UI Behavior - Time field shows current time

- Editable on both platforms

- Format: 24-hour

13.Expected Punch-in Status Punch successful from both devices, recorded with accurate timestamp

Severity :- High:Affects critical user functionality and cross-platform trust.

Priority :- Medium–High

Reason: Must be fixed for mobile parity, but does not block all operations.

Expected Result :-

1.Time field should behave the same across desktop and mobile:

2.Editable if enabled in config.

3.Same default value (e.g., current time).

4.Same format (12h or 24h).

5.Punch-in should be successful on both platforms.

6.No unexpected UI issues or validation differences.

Actual result :-

Pass:

Both desktop and mobile platforms:

1.Show consistent time field behavior.

2.Allow successful punch-in under the same configuration.

3.Display same validations, formats, and default values.

Fail:

1.Time field is editable on one platform but not the other.

2.Mobile app does not default to the correct time.

3.Time format is inconsistent (e.g., 24h on desktop, 12h on mobile).

4.Punch-in fails only on mobile or behaves unexpectedly.

5.UI responsiveness or display issues on mobile only.

------------------------------------------------------------------------------------------------------------------------------------

95.Validate View Daily Attendance Details.

Employee views punch-in, punch-out, total hours, and break time for a date.

Bug Description:- .Any data is missing or incorrect

Time > Attendance > My Records

Steps :-

1.Navigate to Time > Attendance > My Records.

2. Select July 8, 2025 from the calendar/date list.

3. Observe the Punch In, Punch Out, Total Hours, and Break Time fields.

Pre condition :-

1.Employee has valid attendance data for the selected date.

2.User is logged in.

Test Data :-

Employee ID: EMP123

1.Date: 2025-07-08

2.Expected Punch In: 09:00 AM

3.Expected Punch Out: 06:00 PM

4.Break Time: 1 hour

Severity :- Medium – Affects daily productivity insights but not system-wide functions.

Priority :-High – Core functionality expected by all users.

Expected Result :-

1.Punch In = 09:00 AM

2.Punch Out = 06:00 PM

3.Break Time = 1 hour

4.Total Hours = 8 hours (excluding break)

5.All values are correctly formatted (e.g., HH:MM AM/PM).

Actual result :-

Pass:

1.All values (in/out time, total hours, break time) are shown and correctly formatted.

Fail:

1.Any data is missing or incorrect

2.Incorrect time formatting

3.Page crashes or errors on load

------------------------------------------------------------------------------------------------------------------------------------

96.Validate A punch-in exists without a corresponding punch-out.

Bug Description:- Tooltip/message does not appear, is misleading, or is unclear.

Anomaly Flagging (Missed Punch)

Test type :-

1.Test level :Functional Testing

2.Test Type :UI Testing, Validation Testing, Error Handling

3.Testing Phase :System Testing / Integration Testing

Test technique :-

1.Equivalence Partitioning :Group data into valid (complete punch) vs. invalid (missing punch-out).

2.Boundary Value Analysis :Use edge times (e.g., punch-in at 11:59 PM) to ensure anomalies are caught.

3.Error Guessing :Based on likely user mistakes (e.g., forgetting to punch out).

4.Exploratory Testing :Try unexpected data inputs or system states (e.g., duplicate punch-ins).

5.State Transition Testing:Check state changes: normal → incomplete → flagged → corrected.

Steps:-

Step # \_Action

1.Log into the Attendance Management System as a valid user.

2 Navigate to the Attendance Records or Timesheet section.

3 Load a specific date or employee record where a punch-in exists but no punch-out.

4 Observe the row or entry with the missing punch.

5 Hover over or click the ❗ anomaly icon. A tooltip or popup appears with a message like “Missing punch-out time.”

6 Check if the system offers a way to resolve the issue (e.g., edit or notify).

7 (Optional) Attempt to correct the punch-out using the available option.

Pre condition:-

1.User has access to the attendance management system.

2.Attendance data includes at least one entry with a punch-in but no punch-out.

Test Data:-

Employee ID\_ Date\_ Punch-In Time\_ Punch-Out Time\_ Expected Anomaly

1.EMP001\_ 2025-07-08\_ 09:00 AM (Missing) \_ Flag for missing punch-out

2.EMP002 2025-07-08 10:15 AM (Missing)\_ Flag for missing punch-out

3.EMP003 2025-07-08 (Missing) 06:00 PM\_ Flag for missing punch-in

This test specifically covers EMP001 and EMP002 (punch-in exists, punch-out missing).

Severity :-

Severity

Medium

The system functions, but incorrect or incomplete data is flagged and may affect payroll/reporting if not corrected.

Priority:-

High

Timely correction is essential for accurate attendance and salary calculations.

Expected Result :-

1.The system clearly flags the anomaly with a ❗ icon.

2.A message/tool tip explains the issue when interacted with.

3.A correction mechanism (edit, request change, etc.) is available to the user.

Actual Result :-

Pass:

1. Anomaly icon appears next to the attendance row with the missing punch-out.

2.Tooltip or message clearly explains the issue (e.g., "Missing punch-out").

3.User is offered a way to correct or report the issue (edit or notify option).

Fail:

1.No icon appears even though punch-out is missing.

2.Tooltip/message does not appear, is misleading, or is unclear.

3.No correction option is provided to the user.

4.System crashes or fails to load the incomplete punch data.

----------------------------------------------------------------------------------------------------------------------------------

97.Validate Weekly/filtered totals are recalculated based on visible records

Bug Description:- Filter selection does not trigger any UI or data update

Test Type :

1.Functional Testing Ensures filtering logic works as expected.

2.UI Testing Validates that dropdown, calendars, and data refresh correctly.

3.Data Validation Confirms that only relevant records are shown.

4.Regression Testing Ensures changes to date filters don't break total calculation or display behavior.

Technique Description / Use in TC-MR04:

1.Equivalence Partitioning :Group date inputs into valid (within range) and invalid (outside range) categories.

2.Boundary Value Analysis :Test date range edges (e.g., first and last day of week/month).

3.State Transition Testing :System moves from one filtered state to another and updates display/state properly.

4.Error Guessing :Select a range with no data, or overlapping incorrect dates to test robustness.

Test Steps :-

1.Navigate to Attendance Records or Timesheet page

2. Click on the date range filter dropdown

3. Select "Last 7 Days"

4. Select a custom range (e.g., 1st–7th of the current month)

5. Verify that the weekly total hours update accordingly- Totals reflect the new filtered dataset

Precondition:-

Attendance data exists for multiple dates.

Test Data :-

Date Range Selected \_Expected Record Dates \_Expected Weekly Hours (Example)

1.Last 7 Days \_July 3–July 9, 2025\_ Total = 38 hrs (based on entries)

2.Custom: July 1–July 5\_ Only July 1–5 entries visible\_ Total = 24 hrs

Severity :- low -medium

Priority :- medium

Expected Result :-

1.Records displayed match the selected date range only

2.Weekly/filtered totals are recalculated based on visible records

3.UI updates without error or delay

Actual result :-

Pass:

1.Only records within selected range are shown

2.Filter dropdown and date pickers function properly

3.Totals update based on filtered records

4.No errors or data inconsistencies after applying filters

Fail:

1.Records outside selected range are shown

2.Weekly totals do not reflect visible data

3.Filter selection does not trigger any UI or data update

4.Application errors or crashes on filter selection

------------------------------------------------------------------------------------------------------------------------------------

98.Validate System flags attendance anomalies using color and icons.

Bug Description:- .Incorrect colors or mismatched icons vs. legend.

Test Type :-

1.UI Testing :Verify visual consistency and correctness of icons.

2.Accessibility Testing :Ensure icons are perceivable by assistive technologies.

Test technique :-

1.Accessibility Testing :Use screen readers and contrast checkers.

2.Exploratory Testing :Check edge cases, e.g., overlapping anomalies.

Steps:-

1.Open the attendance view for a week containing anomalies.

2 Locate and identify all anomaly icons: (Missed Punch), (Late Punch), (Absent or Critical).

3 Verify color coding and icon usage is consistent across the interface.

4 Check accessibility features (e.g., screen reader labels, contrast).

5 Hover or click on icons to view tooltip or help text.

Pre condition:-

Week attendance data contains anomalies

Test Data :-

Employee ID\_ Date Range \_Anomaly Type\_ Description \_Icon Expected

1.EMP006 \_2025-07-01 to 07-07 \_Missed Punch\_ Punch-in without punch-out

2.EMP007 \_2025-07-02 to 07-07 \_Late Punch \_Punch-in after shift start time

3.EMP008 \_2025-07-05 to 07-07 \_Absent / Critical\_ No punches recorded for the day

4.EMP009 \_2025-07-01 to 07-07 \_Multiple anomalies\_ Combination of missed and late

Severity :- medium-Visual indicators impact user awareness and data clarity; errors could cause misinterpretation but do not break system functionality.

Priority :-Medium -Important for usability and clarity, but doesn’t block core functions. Timely fix is needed to ensure correct user experience.

Expected Result :-

1.All anomaly icons are displayed consistently and correctly.

2.Colors and icons match legend/help text.

3.Icons are accessible (with alt text or ARIA labels).

4.Tooltips provide clear explanations.

Actual result :-

Pass:

1.Icons appear clearly and consistently for each anomaly.

2.Colors and icons align with legend/help text.

3.Icons have proper accessibility support (alt text/ARIA).

4.Tooltips accurately describe the anomalies on hover/click.

Fail:

1.Missing or inconsistent anomaly icons.

2.Incorrect colors or mismatched icons vs. legend.

3.Icons lack accessibility support.

4.Tooltips missing, incorrect, or unclear.

------------------------------------------------------------------------------------------------------------------------------------

99.Validate Calendar View Overlay – Attendance Mapping

Bug Description:- Visual indicators are unclear or not intuitive

Test Type : UI /UX functionality

Test Techniques :-

1) Equivalence Partitioning Grouping date types into classes: workdays, weekends, holidays, and leave. Test one from each to reduce redundancy while ensuring coverage.

2)Boundary Value Analysis Check transition days (e.g., end/start of months, holiday to workday, or weekend to weekday).

3)State Transition Testing Ensure calendar transitions (e.g., switching from one month to another) reflect correct data states.

4)UI Testing Visually confirm overlays are shown accurately and intuitively.

5)Exploratory Testing Navigate freely within the calendar (change views, months, hover actions) to detect UI/UX issues.

6)Error Guessing Test with assumptions like overlapping leaves and holidays, missing attendance data, or long months to catch edge-case bugs.

Test Steps :-

1.Login and navigate to the Calendar View

2 Select a month that contains a mix of workdays, leave, holidays, weekends

3 Observe the calendar overlay and icons

4 Match visual indicators against actual attendance data

5 Hover or click on icons (if applicable) to view details

6 Resize or switch month view to confirm consistency

Pre condition:-

User is logged in

1.User has attendance data for the selected month

2.Calendar overlay feature is enabled

Test Data :-

Sample user attendance data including:

1.Workdays

2.Leave

3.Holidays

4.Weekends

5.Different months to test rollover

Severity :- Medium

Priority :- High

Expected Result :-

1)All dates are visually marked accurately:

1.for leave days

2.for holidays

3. for weekends

2)Normal style for workdays

3)Overlay is clear and intuitive

4)No visual overlap, misalignment, or missing markers

5)Tooltips or popups (if present) display correct date information

Actual Result :-

Pass:

1.All day types are accurately displayed with correct visual markers

2.Calendar is readable and accessible (even with many events)

3.Tooltips/popups (if available) match date metadata

4.Display is responsive and works across date ranges

Fail:

1.Incorrect or missing icons for certain dates

2.Overlay causes visual clutter or misalignment

3.Inaccurate data mapping (e.g., holiday shown as workday)

4.Visual indicators are unclear or not intuitive

------------------------------------------------------------------------------------------------------------------------------------

100.Validate Simulate Overlapping Shifts (Night Shift Spanning Two Days)

Cross day attendance

Bug Description:- Shift is split across two entries without config allowing that Hours shown are incorrect (e.g., only partial duration counted)

1.Module :-Time and attendance -My Records

2.Test type :- Test Type Functional + Data Integrity

3.Test Technique :-Boundary Value Analysis, State Transition, Error Guessing

Test Steps :-

1.Log in as an employee or admin

2 Manually or automatically log attendance:

IN: July 10, 10:00 PM

OUT: July 11, 6:00 AM

3 Navigate to My Records page

4 Verify the entire shift duration is recorded as a single, continuous entry

5 Check calculated total hours worked, and ensure summary is accurate

6 View daily summary for July 10 and July 11

7 Optionally export data or query database for backend validation

Pre condition:-

1.Shift policy is configured to allow cross-midnight shifts

2.User has access to log hours (or automated punch-in/out available)

3.No existing entries for that user on the test date

4.System supports multi-day time blocks without splitting into separate records by default

Test Data :-

Field \_Value

1.Employee ID \_EMP\_1023

2.Shift Start \_10:00 PM (Day 1)

3.Shift End\_ 06:00 AM (Day 2)

4.Date \_July 10 (10:00 PM) – July 11 (06:00 AM)

5.Config\_ Night shift policy allowing cross-day work hours (e.g., shift span = 8 hrs)

6.Total duration: 8 hrs

Severity :- High

Priority :- High

ExpectedResult :-

1.Shift is shown as one complete record (10 PM – 6 AM = 8 hours)

2.No unintended split or duplication of records across midnight

3.Daily/weekly summary reflects correct total hours

4.UI shows shift on either the start day or linked both days with correct reference

5.Exported or queried data retains accurate timestamps and duration

Actual Result :-

Pass:

1.Shift is not split at midnight

2.Duration = 8 hours is shown correctly

3.No overlap errors, misalignment, or date mismatch

4.Weekly and daily summary reflect correct totals

Fail:

1.Shift is split across two entries without config allowing that

2.Hours shown are incorrect (e.g., only partial duration counted)

3.Record appears under wrong day or not at all

4.UI displays “invalid shift time” or causes sync error

------------------------------------------------------------------------------------------------------------------------------------

101.Validate Query Incomplete Punch Records (Punch-In Exists but Punch-Out is Null)

Bug Description:- Records missing from results, extra records included, or query errors occur

(Returns incomplete records; used to flag missed punches)

Module :Time & Attendance – Data Validation / Audit

Test type : SQL Query / Data Validation

Steps :-

1.Connect to the attendance database

2 Execute the query:

SELECT employee\_id, punch\_in\_time, punch\_out\_time

FROM attendance\_records

WHERE punch\_in\_time IS NOT NULL

AND punch\_out\_time IS NULL;

3 Review the query results

4 Verify that only records with missing punch-out time are returned

Pre condition :-

1.Attendance records exist in the database

2.Table contains punch\_in\_time and punch\_out\_time fields

3.Some records have punch-in time but missing punch-out time

Test Data :-

employee\_id :punch\_in\_time: punch\_out\_time

1.EMP001\_ 2025-07-10 09:00:00\_ NULL

2.EMP002 \_2025-07-10 08:45:00 \_2025-07-10 17:00:00

3.EMP003 \_2025-07-10 09:30:00 \_NULL

Severity :- medium

Priority :- High

Expected Result :-

1.Query returns all records where punch\_in\_time is present but punch\_out\_time is NULL

2.These records correctly represent incomplete punch data indicating missed punche

Actual Result :-

Pass: Returned records match test data incomplete punches exactly

Fail : Records missing from results, extra records included, or query errors occur

------------------------------------------------------------------------------------------------------------------------------------

102.Validate system prevents duplicate punch entries via DB constraints or logic

Bug Description:- 1.Query returns 2 or more identical records

2.No constraint or validation in place to block duplicates

Module Time & Attendance – Data Integrity / Punch Handling

1.Test Type :SQL Validation, Data Integrity Testing

2.Test Technique: Error Guessing, Negative Testing, Equivalence Partitioning

Steps :-

1.Insert a valid punch for user 123 at '2025-07-12 09:00:00'

2 Attempt to insert a second punch with the same user\_id and timestamp

3 Run the SQL query:

SELECT \* FROM attendance WHERE user\_id = 123 AND timestamp = '2025-07-12 09:00:00';

4 Check how many rows are returned

Pre condition :-

1)The attendance table contains time punch records with at least the following fields:

1.user\_id

2.timestamp (datetime of punch)

2)Database should either:

1.Enforce a unique constraint on (user\_id, timestamp), or

2.Backend should check and prevent duplicate submissions

Test Data :-

user\_id \_timestamp

1)123 \_2025-07-12 09:00:00

Severity :- High

Priority :- High

Expected Result :-

1.Query returns only one record

2.Second insert is blocked by either:

3.A unique constraint violation error, or

4.Backend validation logic

5.No duplicate rows exist for the same user/timestamp combination

Actual result :-

Pass:

1.Duplicate punch attempt is not allowed

2.Only one record exists for a given (user\_id, timestamp)

3.System provides a clear error message or silently prevents insert

4.Data integrity is preserved in the table

Fail:

1.Duplicate punches are allowed for the same timestamp and user

2.Query returns 2 or more identical records

3.No constraint or validation in place to block duplicates

4.System accepts and stores redundant or corrupt data

------------------------------------------------------------------------------------------------------------------------------------

1. Validate group rule application using JOIN on user\_group\_config

Bug description:- Overtime calculated for users where it's disallowed, or vice versa

Module Time & Attendance – Group Configuration / Rule Mapping

1.Test Type :SQL Join Logic, Data Validation, Rule Consistency

2.Test Technique :Join Testing, Equivalence Partitioning, Rule-Based Testing

Steps :-

SELECT

a.user\_id,

a.punch\_in\_time,

a.punch\_out\_time,

g.group\_name,

g.work\_hours\_required,

g.allow\_overtime

FROM

attendance a

JOIN

user\_group\_config g ON a.group\_id = g.group\_id;

1.Run the JOIN query to fetch users' attendance with their group rules

2 Check if users' group rules are correctly joined and reflected in result

3 Verify exports or processing (e.g., overtime calculation) aligns with fetched group config

4 Cross-check one user from each group for expected behavior

5 For edge cases (e.g., user with multiple groups), verify the highest priority rule is applied

Pre condition :-

Group Rule Element Examples:

1.Required Work Hours Full-time: 8 hours/day

2.Part-time: 4 hours/day

3.Grace Period 15 minutes late tolerance for full-time

4.Overtime Eligibility Contractors: No OT

5.Full-time: OT after 9 hours

6.Shift Type Fixed, Rotational, Flexible

7.Punch Rounding Rules Round to nearest 5/10 minutes

8.Leave and Holiday Policies Full-time get paid leave, interns don’t

1.attendance table contains user attendance data

2.user\_group\_config table defines group-specific rules (e.g., grace periods, working hours, OT policy)

3.Each user has a valid group\_id linked to rules

4.Export or processing logic references group config

Test data :-

user\_id ,group\_id, group\_name ,work\_hours\_required ,allow\_overtime

1.U001 G1 Full-Time 8.0 Yes

2.U002 G2 Contractor 4.0 No

Severity:- High

Priority :- High

Expected Result :- 1.Each record shows the correct group rule (e.g., Full-Time, Contractor)

2.Exported data (e.g., total required hours, overtime eligibility) reflects those rules

3.No mismatch between user group and applied rules

4.For users in multiple groups, rule precedence is respected (if defined)

Actual Result :-

Pass:

1.SQL JOIN returns correct group details per user

2.Processed/exported data aligns with rules from user\_group\_config

3.Attendance is calculated and validated based on correct group policy

4.No missing or incorrect rule applications

Fail:

1.Group rule mismatches for users

2.Attendance exported with wrong thresholds or policies

3.Missing or incorrect group\_id associations

4.Overtime calculated for users where it's disallowed, or vice versa

------------------------------------------------------------------------------------------------------------------------------------

104.Validate Punch In/Out from a Different Time Zone

Bug Description:- Shift association, overtime, or leave triggers incorrectly

(Punch from Different Time Zone – Normalize to Base Time Zone)

Module: Time & Attendance – Time Zone Handling

1.Test Type :Functional, Internationalization, Time Logic

2.Test Technique : Equivalence Partitioning, Time Simulation, Environment Switching

Steps :-

1.Log in as user 115 from a device set to PDT (UTC-7)

2 Perform Punch In at 08:00 local time

3 Perform Punch Out at 16:00 local time

4 View the punch times in My Records, Admin View, and database

5 Confirm that punch times are:

- Stored and shown in home time zone (IST) or UTC, not in local client time

- Clearly labeled with time zone if not normalized

- Accurate and aligned across all views

SELECT punch\_time AT TIME ZONE 'UTC' AT TIME ZONE 'Asia/Kolkata' AS punch\_time\_ist

FROM attendance

WHERE user\_id = 115

AND punch\_time::date = '2025-07-12';

Expected: 2025-07-12 20:30:00 IST

Pre condition :-

1.User's home location and default time zone are set (e.g., IST or UTC+5:30)

2.User has mobile/web access to punch remotely

3.System supports time zone-aware entries or uses server-side normalization

Test data :-

Field\_ Value

1.User ID\_ 115

2.Home Time Zone \_UTC+05:30 (IST)

3.Current Time Zone\_ UTC-07:00 (PDT – Pacific Time, U.S.)

4.Punch In (Local) \_2025-07-12 08:00 PDT

5.Punch In (Home TZ)\_ 2025-07-12 20:30 IST

Severity :- High

Priority :- High

Expected Result :-

1.Punch times are normalized to IST (UTC+5:30) or stored in UTC internally

2.UI clearly displays times according to:

3.User’s base time zone (or as per org policy)

Or includes time zone info if not normalized

4.No duplicate or misaligned punches due to time difference

5.Duration between punch-in/out is accurate (8 hours)

Actual result :-

Punch-in saved as 2025-07-12 20:30 IST

1. UI shows normalized time with label: 20:30 IST

2. Duration: 8h recorded correctly

3. Result: PASS

Pass:

1.System correctly converts and stores times regardless of device/client time zone

2.Times are uniform across reports, summaries, and exports

3.No confusion in shift mapping, overtime, or calendar view

4.Optional: time zone shown for clarity in audit logs

Fail:

1.Times are stored in client local time without conversion

2.UI shows wrong day due to time zone misalignment

3.Shift association, overtime, or leave triggers incorrectly

4.Admin/user sees inconsistent punch timestamps

------------------------------------------------------------------------------------------------------------------------------------

105.Validate Generate report for a week/month

Bug Descrption:- Missing entries within the range

Module :Time Reports / Project Reports

Test Type: Functional, Date Range Filtering

Steps:-

1. Log in and go to Reports module.

2. Select:

 • Project = "Client Portal Upgrade"

 • User = "Sofia Khan"

 • Date Range = 01–07 July 2025 (or 01–31 July for monthly test)

3. Click Generate Report.

4. Review time entries shown in the report.

Pre condition :-

Time entries exist for multiple dates within and outside the selected range

Test data :-

1.Project: “Client Portal Upgrade”

2.User: “Sofia Khan”

3.Date Range: 01–07 July 2025 (week) or 01–31 July 2025 (month)

4.Entries: Logged on 02, 04, 10, 15, 22 July

Severity :- High

Priority :- High

Expected result :-

1.Report displays only time entries within the selected date range.

2.No entries before 01 July or after 07 July (or 31 July for month) are included.

3.Daily and total hours are correctly calculated based on in-range entries only.

Actual result :-

Pass:

Report includes only:

 1. Dates within the selected range

 2.Time entries correctly filtered

 3. Correct totals per day and overall

 4. Accurate user and project info

Fail:

Report shows:

 1.Entries outside of selected date range

 2.Missing entries within the range

 3.Totals include out-of-range hours

 4.Report incorrectly includes other users or projects

------------------------------------------------------------------------------------------------------------------------------------

106.Validate Future date selected-No data or appropriate message

Module :Time Reports / Project Reports

Bug description:- No unexpected behavior (e.g., crash, broken UI)

Test Type :Negative Testing, Boundary Testing

Test Steps :-

1. Navigate to the Reports module.

2. Select a valid project and user (optional).

3. Set the date range to 20–31 August 2025 (future period).

4. Click Generate Report.

Pre condition:-

No time entries are logged for future dates

Test data :-

1. Start Date: 20 August 2025

2.End Date: 31 August 2025 (future range)

3.Current Date: 13 July 2025

Severity :- medium

Priority :- medium

Expected Result :-

1) The system should show:

 1. A message like “No data available for selected date range”, or

 2.An empty report grid with proper headings but no data rows

1. No error or crash should occur

Actual Result :-

Pass:

The system handles the future date range gracefully

1. A clear message or empty report is shown

2. No unexpected behavior (e.g., crash, broken UI)

------------------------------------------------------------------------------------------------------------------------------------

107.Validate No data for selected filters – “No data available” message shown

Bug Description:- Message is unclear (e.g., “null”, “undefined”)

1.Module :Time Reports / Project Reports

2.Test Type :Negative Testing, UX/Feedback

Test steps :-

1. Navigate to the Reports section.

2. Apply filters that yield no data:

 • Select a user/project with no entries

 • Choose future or empty date range

3. Click Generate Report.

4. Observe the result panel.

Pre condition:-

At least one filter combination must return no data (e.g., future dates, unassigned users)

Test data :-

1.User: Carol (no time logged)

2. Date Range: Future (e.g., Jan 1–10, 2026)

3. Project: Unused project

Severity :- medium

Priority :- High

Expected Result :-

1. System displays a clear, friendly message, such as:

 > “No data available for the selected filters.”

2. No errors, broken layout, or blank screen

3. Optionally suggest checking filters or changing date range

Actual Result :-

Pass:

1.System shows a helpful “No data available” message

2. No UI elements break or disappear

3. Report panel remains consistent in layout

4. Export buttons are disabled or hidden appropriately

Fail :

1.Report area is blank with no message

2. UI crashes or throws error

3. Message is unclear (e.g., “null”, “undefined”)

4. Export option is active despite no data

------------------------------------------------------------------------------------------------------------------------------------

1. Validate Simultaneous requests (multi-user) – System handles concurrency properly

Module: Time Reports / Project Reports – Performance / Stability

Test Type: Load Testing, Performance, Concurrency

Bug Description:- Data from one user appears in another user’s report

Test steps :-

1. Have 5–10 users simultaneously generate reports for similar filters.

 • You may simulate this using browser sessions, load testing tools (e.g., JMeter, Postman Runner), or real users.

2. Observe:

 • Response time

 • Server load

 • UI behavior

 • Data correctness

3. Monitor for race conditions, UI freezes, or API errors.

4. Validate that results match expected data and no requests are dropped.

Pre condition:-

1.Multiple users have access to generate reports

2. Time entries exist for shared or overlapping periods

Test data :-

1. 5+ users with report access (Admin, Manager, Employees)

2. Logged time data for same projects/date ranges

3. Example filters:

 • Date Range: July 1–7, 2025

 • Project: "Marketing Launch"

Severity :- High

Priorty :- High

Expected Result :-

1.All users receive complete and correct reports

2. No timeouts, crashes, or duplicate/missing entries

3. Server manages load efficiently without slowing down dramatically

4. UI remains responsive and consistent for all users

Expected Result :-

Pass:

1.System handles concurrent report generation without error

2. Response time remains acceptable

3. No overlap, data corruption, or missing fields

4. Each user sees the correct result

Fail:

1.API fails under load (500/504 errors)

2. Reports are incomplete or inconsistent

3. UI freezes, crashes, or shows wrong data

4. Data from one user appears in another user’s report

------------------------------------------------------------------------------------------------------------------------------------

109.Validate total hours per user – Check that SQL query correctly sums hours logged per user

(Ensure SQL correctly aggregates and sums time logged per user over a selected time period.)

Bug Description:- Missing users with valid data

Module :-Reports / Project Reports – Backend SQL

Test Type :Functional, Data Accuracy, SQL Validation

Steps :-

1. Run the SQL query above.

2. Compare each user’s total with manual calculation from individual records.

3. Optionally, run individual queries per user to verify:

 SELECT SUM(hours\_logged) FROM time\_entries WHERE user\_id = 'user\_a';

1. Validate returned totals are accurate.

Pre condition:-

SELECT user\_id, SUM(hours\_logged) AS total\_hours

FROM time\_entries

WHERE date BETWEEN '2025-07-01' AND '2025-07-07'

GROUP BY user\_id;

Test data:-

Example data:

 • User A: 3 entries (2h, 4h, 1.5h)

 • User B: 2 entries (3h, 2h)

 • User C: 0 entries

Severity :High

Priority :High

Expected Result :-

Pass:

1.Each user appears once with the total number of hours they logged.

2.Users with no time entries may be excluded unless joined from a users table.

3. Sums are mathematically correct.

Fail:

1.Incorrect totals

2. Missing users with valid data

3. SQL error or poor performance on large datasets

ACtual result :-

Pass:

1.Totals are accurate and match expected values

2. No data is inflated or missed

3. Query completes successfully and efficiently

Fail:

1.Incorrect totals

2. Missing users with valid data

3. SQL error or poor performance on large datasets

------------------------------------------------------------------------------------------------------------------------------------

110.Validate Date range filter accuracy – Verify WHERE date BETWEEN start AND end returns only relevant rows

(Ensure SQL queries using WHERE date BETWEEN start AND end return only records within the specified date range.)

Bug Description:- Boundary dates (start/end) are excluded

Module: Reports / Project Reports – SQL Filtering

Date range filter accuracy

Test Type :Functional, SQL Validation, Data Filtering

Steps :-

1. Insert or identify test data spanning before, within, and after the range.

2. Run the above SQL query.

3. Verify that:

 • Only entries dated July 1 to July 5 are returned

 • No entries from June 30 or July 10 are included

1. Confirm edge dates (start and end) are included.

Pre condition:-

SELECT \*

FROM time\_entries

WHERE date BETWEEN '2025-07-01' AND '2025-07-05';<br>

1.time\_entries table contains entries across multiple dates.

2. Column date is in valid DATE format.

Test data :-

Sample dataset:

 1. User A logs: July 1, 2, 3

 2. User B logs: June 30, July 4, July 10

 3. Date Range tested: 2025-07-01 to 2025-07-05

Severity : medium

Priority :- high

Expected Result :- 1.The query returns only records where date is between '2025-07-01' and '2025-07-05' inclusive.

2. No rows outside this range should appear.

3. The date boundaries are respected properly.

Actual result :-

Pass:

1.Only rows within the specified date range are returned

2. Start and end dates are included

3. Query works consistently across different date ranges

Fail:

1.Records outside the date range are returned

2. Boundary dates (start/end) are excluded

3. Query fails due to data type mismatch or wrong format

------------------------------------------------------------------------------------------------------------------------------------

111.Validate updates to projects with status = 'Archived' are blocked or rejected

Module :Time\_ProjectInfo\_Projects

Bug Description:- Update succeeds despite status = 'Archived'

Steps :-

1.Insert a test project with status = 'Archived' Insert is successful

2 Attempt to update fields like project\_name, status, or end\_date Update fails or is blocked

3 Observe the error or validation message returned Message like: Modification not allowed on archived projects or SQL error

4 Try update from API or UI (if applicable) Application shows a validation or access restriction

5 Ensure that read-only enforcement applies across all entry points (UI/API/DB) Consistent enforcement

Pre condition :-

Update Attempt (should fail):

UPDATE Time\_ProjectInfo\_Projects

SET project\_name = 'Archived Project Renamed'

WHERE project\_id = 701;

Test data :-

project\_id :project\_code :project\_name :status :start\_date :end\_date

701 PRJ-701 Old CRM System Archived 2024-01-01 2024-06-01

Severity :- High

Priority :- High

Expected Result :-

Any attempt to modify an archived project fails (either at DB, API, or UI level)

Actual result :-

Pass:

1.Updates to archived projects are rejected

2.Clear and meaningful error or restriction is provided

3.Record remains unchanged

4.Behavior is consistent across interfaces

Fail:

1.Update succeeds despite status = 'Archived'

2.No restriction exists at DB, API, or UI level

3.Record is silently modified

4.Different behavior across UI/API/DB

------------------------------------------------------------------------------------------------------------------------------------

112.Validate the system detects users logging overlapping time entries on multiple projects on the same day.

Bug Description:- Overlaps missed or false positives returned.

Steps :-

SELECT

e1.user\_id,

e1.entry\_date,

e1.project\_id AS proj1,

e2.project\_id AS proj2,

e1.start\_time AS start1,

e1.end\_time AS end1,

e2.start\_time AS start2,

e2.end\_time AS end2

FROM Time\_Entries e1

JOIN Time\_Entries e2

ON e1.user\_id = e2.user\_id

AND e1.entry\_date = e2.entry\_date

AND e1.project\_id <> e2.project\_id

AND e1.id <> e2.id

AND e1.start\_time < e2.end\_time

AND e1.end\_time > e2.start\_time;

1.Insert two overlapping time entries for the same user, on the same date, but different projects (e.g., 10:00–11:00 and 10:30–11:30).

2.Insert a non-overlapping entry as control (e.g., 12:00–13:00).

3.Run the query.

4.Check that only overlapping entries are returned.

Pre condition :-

The Time\_Entries table must allow overlapping time logs and store start and end times in a time or datetime format.

Test data :-

User: 200

1.Project A: 10:00–11:00

2.Project B: 10:30–11:30 (same day)

3.Project C: 12:00–13:00 (no overlap)

Severity :- Critical – Could result in billing fraud or compliance violations

Priority :- High – Time integrity must be preserved

Expected Result :- The overlapping entries (Project A & B) for User 200 are returned by the query.

Actual result:- 1.Pass Criteria: Query returns overlapping project entries.

1. Fail Criteria: Overlaps missed or false positives returned.

------------------------------------------------------------------------------------------------------------------------------------

113.Validate Entry With Invalid Characters in Notes

Bug Description:- Emoji/HTML breaks frontend rendering

(Validate that the notes field sanitizes or rejects unsafe content like emojis, HTML, or SQL)

Module :Time\_Entries, Input Validation, Sanitization Logic

Steps :-

1.Enter time with normal note ("Worked on feature A") Accepted

2 Enter note with <script> tag Rejected or script is removed

3 Enter SQL-like string ('; DROP TABLE ...) Rejected or stored safely (escaped)

4 Enter note with emojis ("Good job!") Accepted or cleaned based on config

5 Try very long input (e.g., 10,000 characters) Rejected if above max allowed length (e.g., 1000 chars)

6 Verify DB doesn't store harmful content Notes field is clean and secure

SQL Insert (Unsafe):-

INSERT INTO Time\_Entries (

user\_id, project\_id, entry\_date, hours\_logged, notes

) VALUES (

'U009', 'P7001', '2025-07-14', 2.0, '<script>alert("XSS")</script>'

);

Pre condition :-

1.Time entry UI/API allows free-text notes field

2.System should guard against:

1) SQL injection

2) HTML tags/scripts

3) Emojis and unsupported Unicode

SQL Insert (Unsafe):-

INSERT INTO Time\_Entries (

user\_id, project\_id, entry\_date, hours\_logged, notes

) VALUES (

'U009', 'P7001', '2025-07-14', 2.0, '<script>alert("XSS")</script>'

);

Test Data :-

Input Type :Sample Note :Expected Behavior

1. Valid Text "Meeting with client" Accepted

2. HTML Tag "<script>alert(1)</script>" Rejected or sanitized

3. SQL Injection "'; DROP TABLE Time\_Entries; --" Rejected or sanitized

4. Emoji Input "Finished work" Accepted (if allowed) or removed

5. Excess Length 5000 characters of "A" Rejected (if limit = 1000)

Severity : medium

Priority : High

Expected Result :- Notes are either sanitized (stripped of unsafe characters) or rejected with a message

Actual result :- Pass:

1.Unsafe input is not stored or is stored safely

2.HTML/script tags are removed or escaped

3.SQL injection content doesn't execute or store as-is

4.Emoji/text is preserved or cleaned as per requirements

5.Input length is enforced

Fail:

1.Scripts or SQL injected into DB

2.Emoji/HTML breaks frontend rendering

3.Notes field allows unsafe or excessive input

------------------------------------------------------------------------------------------------------------------------------------

114.Validate Ensure that no time entries can be created or modified for projects with status 'Archived'

Bug description:- Time entries are allowed or processed for archived projects

Module :- Time\_Entries, Time\_ProjectInfo\_Projects, Time Logging UI

Test steps :-

1.Identify or create a project with status = 'Archived' Project is archived

2 Attempt to log time via UI or API for the archived project System prevents the action

3 Check for validation message User receives clear error: “Cannot log time to an archived project”

4 Try bulk import of time including archived project entries Rows linked to archived projects are skipped or marked invalid

5 Attempt to edit a past time entry that was originally logged to the now-archived project Update is blocked unless explicitly allowed (e.g., by Admin override)

Pre condition :-

1.A project exists with status = 'Archived'

2.The time logging feature (UI or API) references project\_id

3.Status field is respected in business rules during inserts/updates

Test data :-

Project ID: Name :Status :User ID: Time Entry Date :Hours :Expected Result

1.P5001 Legacy App Archived U101 2025-07-14 2.0 Rejected

Severity :High

Priority :High

Expected Result :- Time logging is rejected or blocked if the project is marked as Archived

Actual Result :- Pass:

1.System blocks time entry creation or updates for archived projects

2.Proper error messages are displayed in UI or returned in API

3.Bulk imports validate against project status

4.Archived projects do not appear in time-entry dropdowns/selections

Fail:

1.Time entries are allowed or processed for archived projects

2.UI/API allows time logging without checking project status

3.Archived project appears in active project selection lists

------------------------------------------------------------------------------------------------------------------------------------

115.Validate Ensure that all time entries have positive hours logged.

Entries with hours\_logged <= 0 should be flagged or rejected.

Bug Description:- uery returns any row with hours\_logged <= 0 Fail — Invalid data present

Steps :-

1.Insert time entries with positive, zero, and negative hours Mixed data in Time\_Entries

2 Execute the SQL query above Returns only invalid records (hours ≤ 0)

3 Confirm entries with 0 or < 0 hours appear in the result Matches expected invalid entries

4 Fix the bad data and re-run the query Query returns 0 rows → Pass

SELECT \* FROM Time\_Entries

WHERE hours\_logged <= 0;

Pre condition :-

SELECT \*

FROM Time\_Entries

WHERE hours\_logged <= 0;

Test data :-

entry\_id: user\_id: project\_id :entry\_date: hours\_logged

1 U001 101 2025-07-10 8

2 U002 101 2025-07-10 0

3 U003 102 2025-07-11 -2

4 U004 103 2025-07-11 4

Severity :- High

Priority :High

Expected Result :-

Criteria \_Expectation

1. Pass Query returns zero rows → all time entries are valid (hours > 0)

2. Fail Query returns any row → system contains invalid entries with 0 or negative hours

Example Failing Output:

entry\_id user\_id hours\_logged

102 U002 0

103 U003 -3

Actual result :-

Pass:-

1.Query returns 0 rows Pass — No invalid entries

2.All values of hours\_logged are strictly greater than 0 Pass — Data is clean

3.System enforces validation on insert/update preventing ≤ 0 values Pass — Rules properly applied

Fail:-

1.Condition Result

2.Query returns any row with hours\_logged <= 0 Fail — Invalid data present

3.At least one entry has 0 or negative hours Fail — Requires correction

4.System allows saving such entries without warnings/errors Fail — Validation missing

------------------------------------------------------------------------------------------------------------------------------------

1. Validate the system correctly identifies time entries logged on weekends (Saturday and Sunday).

Bug Description:- Weekday entries are included or weekend entries are missing.

Test Steps :-

SELECT \*

FROM Time\_Entries

WHERE EXTRACT(DOW FROM entry\_date) IN (0, 6);

SELECT \*

FROM Time\_Entries

WHERE DATEPART(WEEKDAY, entry\_date) IN (1, 7); -- May need SET DATEFIRST

SQL Server WEEKDAY values depend on SET DATEFIRST. By default, 7 = Saturday, 1 = Sunday (U.S. default).

1.Login as a valid user.

2.Navigate to the time entry screen or reporting tool.

3.Submit or simulate time entries with different entry\_date values (including weekdays and weekends).

4.Run the query/report: SELECT \* FROM Time\_Entries WHERE EXTRACT(DOW FROM entry\_date) IN (0, 6);

5.Verify that only entries from Saturday (6) and Sunday (0) are returned.

Pre condition :-

1.Time entries must exist in the Time\_Entries table.

2.At least some entries should be from weekends.

Test data :-

1.Entry 1: 2025-07-12 (Saturday)

2.Entry 2: 2025-07-13 (Sunday)

3.Entry 3: 2025-07-14 (Monday)

Severity :-Medium – Primarily used for auditing or policy compliance

Priority :- Medium – Important for time tracking accuracy and potential policy enforcement

Expected Result :-

The query/report should return only entries logged on Saturday or Sunday (Entry 1 and Entry 2 in the test data).

Actual result :-

1.Pass Criteria: Only weekend entries appear in the result.

2. Fail Criteria: Weekday entries are included or weekend entries are missing.

------------------------------------------------------------------------------------------------------------------------------------