Login Features							
Test Case ID	**Title**	**Scenario**	**Steps**	**Expected Result**	**Actual Result**	**Severity**	**Priority**
TC_LOGIN_01	Verify that user can login with valid credentials	Valid login	1. Open the application login page. 2. Enter a valid registered email address. 3. Enter the correct password. 4. Click on the Login button.	User should be authenticated successfully and redirected to the Dashboard page with an active session created.	To be executed	Critical	High
TC_LOGIN_02	Verify that system shows error for invalid password	Invalid password	1. Open the login page. 2. Enter a valid email address. 3. Enter an incorrect password. 4. Click on Login.	System should deny login and display a clear error message indicating invalid credentials.	To be executed	Major	High
TC_LOGIN_03	Verify that system validates incorrect email format	Invalid email format	1. Open the login page. 2. Enter email in invalid format (e.g. user@com). 3. Enter any password. 4. Click on Login.	System should not allow login and should display email format validation message.	To be executed	Major	High
TC_LOGIN_04	Verify that system does not allow login with empty fields	Empty input	1. Open the login page. 2. Leave email field empty. 3. Leave password field empty. 4. Click on Login.	System should block login and display required field validation for both email and password.	To be executed	Major	High
TC_LOGIN_05	Verify that password is mandatory for login	Missing password	1. Open the login page. 2. Enter valid email address. 3. Leave password field empty. 4. Click on Login.	System should block login and display validation message for missing password.	To be executed	Major	High
TC_LOGIN_06	Verify that email field is case-insensitive	Uppercase email	1. Open the login page. 2. Enter valid email in uppercase letters. 3. Enter correct password. 4. Click on Login.	System should successfully authenticate user regardless of email letter case.	To be executed	Minor	Medium
TC_LOGIN_07	Verify that system handles very long input values	Long values	1. Open the login page. 2. Enter more than 200 characters in email field. 3. Enter more than 200 characters in password field. 4. Click on Login.	System should restrict input length or show validation without crashing or freezing.	To be executed	Major	Medium
TC_LOGIN_08	Verify that system is protected against script/SQL injection	Security input	1. Open the login page. 2. Enter ' OR 1=1 -- in email field. 3. Enter same in password field. 4. Click on Login.	System should sanitize inputs, prevent login, and should not expose any technical error.	To be executed	Critical	High
TC_LOGIN_09	Verify that system handles multiple failed login attempts	Brute force	1. Open the login page. 2. Enter valid email. 3. Enter wrong password five times. 4. Click on Login each time.	System should remain stable and continue showing appropriate error messages without crash.	To be executed	Major	Medium
TC_LOGIN_10	Verify that user session persists after browser refresh	Session check	1. Login with valid credentials. 2. Once on dashboard, refresh the browser.	User should remain logged in and stay on Dashboard with active session.	To be executed	Critical	High
Dashboard Feature							
Test Case ID	**Title**	**Scenario**	**Steps**	**Expected Result**	**Actual Result**	**Severity**	**Priority**
TC_DASH_01	Verify that dashboard loads successfully after login	Normal load	1. Open login page. 2. Login with valid credentials.	Dashboard should load without errors and display all required components.	To be executed	Critical	High
TC_DASH_02	Verify that correct user information is displayed	Data validation	1. Login to application. 2. Observe user name and email on dashboard.	Dashboard should display user details matching logged-in account.	To be executed	Major	High
TC_DASH_03	Verify that recent activity is displayed correctly	Activity display	1. Login to application. 2. Perform any action (upload file or update data). 3. Navigate to dashboard.	Dashboard should update and show latest user activity.	To be executed	Major	Medium
TC_DASH_04	Verify that system shows proper empty state for new users	New user	1. Login using a new user account with no data.	Dashboard should show meaningful empty state message instead of blank screen.	To be executed	Minor	Low

TC_DASH_05	Verify that unauthorized users cannot access dashboard	Security	1. Open browser in incognito mode. 2. Directly access dashboard URL.	System should redirect user to Login page.	To be executed	Critical	High
TC_DASH_06	Verify that system handles backend failure gracefully	API error	1. Login to application. 2. Simulate API failure (disconnect network or mock). 3. Refresh dashboard.	Dashboard should show user-friendly error message without crashing UI.	To be executed	Major	High
TC_DASH_07	Verify that dashboard reloads correctly on page refresh	Page reload	1. Login to application. 2. Refresh dashboard page.	Dashboard data should reload correctly without loss of session.	To be executed	Minor	Medium
TC_DASH_08	Verify that dashboard UI is responsive on smaller screens	UI test	1. Login to application. 2. Resize browser window to mobile size.	Dashboard layout should adjust properly without overlapping or breaking.	To be executed	Minor	Low
FILE UPLOAD							
Test Case ID	**Title**	**Scenario**	**Steps**	**Expected Result**	**Actual Result**	**Severity**	**Priority**
TC_FILE_01	Verify that user can upload a valid PDF file	Happy path	1. Login to application. 2. Navigate to File Upload section. 3. Click on Upload button. 4. Select a PDF file under 10MB. 5. Click Submit.	System should upload the file successfully and display confirmation message.	To be executed	Major	High
TC_FILE_02	Verify that user can upload a valid image file	Happy path	1. Login to application. 2. Navigate to File Upload section. 3. Click Upload. 4. Select JPG/PNG file under 10MB. 5. Click Submit.	System should upload image successfully and display preview if supported.	To be executed	Major	High
TC_FILE_03	Verify that system restricts file size above limit	Size validation	1. Login to application. 2. Navigate to upload section. 3. Select file greater than 10MB. 4. Click Submit.	System should reject file and display file size limit error message.	To be executed	Critical	High
TC_FILE_04	Verify that system blocks unsupported file types	Type validation	1. Login to application. 2. Navigate to upload section. 3. Select unsupported file (e.g. .exe). 4. Click Submit.	System should reject file and display unsupported file type error.	To be executed	Critical	High
TC_FILE_05	Verify that system does not allow upload without selecting file	Empty input	1. Login to application. 2. Navigate to upload section. 3. Click Submit without selecting file.	System should show validation message asking user to select a file.	To be executed	Minor	Medium
TC_FILE_06	Verify that system handles corrupted files properly	File integrity	1. Login to application. 2. Navigate to upload section. 3. Select a corrupted PDF/image file. 4. Click Submit.	System should fail upload and display appropriate error message.	To be executed	Major	Medium
TC_FILE_07	Verify that system handles duplicate file uploads	Re-upload	1. Login to application. 2. Upload a valid file successfully. 3. Upload the same file again.	System should either replace file or notify user without error.	To be executed	Minor	Low
TC_FILE_08	Verify that user can cancel file upload	Cancel flow	1. Login to application. 2. Start uploading a large file. 3. Click Cancel button.	System should stop upload process without errors.	To be executed	Minor	Low
TC_FILE_09	Verify that system handles slow network during upload	Network issue	1. Login to application. 2. Enable slow network mode. 3. Upload file.	System should show loader/progress without freezing UI.	To be executed	Major	Medium
TC_FILE_10	Verify that upload process stops on page refresh	Interrupt	1. Login to application. 2. Start uploading file. 3. Refresh browser during upload.	Upload should be cancelled safely without data corruption.	To be executed	Major	Medium
LOGOUT							
Test Case ID	**Title**	**Scenario**	**Steps**	**Expected Result**	**Actual Result**	**Severity**	**Priority**
TC_LOGOUT_01	Verify that user can logout successfully	Happy path	1. Login to application. 2. Click on Logout button.	User should be logged out and redirected to Login page.	To be executed	Major	High
TC_LOGOUT_02	Verify that user cannot access dashboard after logout	Security	1. Logout from application. 2. Click browser Back button.	System should prevent access and redirect user to Login page.	To be executed	Critical	High
TC_LOGOUT_03	Verify that session ends after logout and refresh	Session end	1. Logout from application. 2. Refresh browser.	User should remain logged out with no active session.	To be executed	Major	High

TC_LOGOUT_04	Verify that logout works across multiple tabs	Session sync	1. Login in two browser tabs. 2. Logout from one tab. 3. Refresh second tab.	User should be logged out from all active tabs.	To be executed	Critical	High
TC_LOGOUT_05	Verify that user is logged out after session timeout	Idle timeout	1. Login to application. 2. Stay idle until session expires. 3. Refresh page.	System should automatically log out user and redirect to Login page.	To be executed	Major	Medium
High-Risk Areas to Focus on Before Release							
Risk ID	**High-Risk Area**	**Description**	**Why This Is High Risk**	**Business Impact**			
1	Authentication & Session Management	This area covers user login, credential validation, session creation, and session persistence across the application.	1. It is the primary entry point for all users. 2. Any failure can completely block users from accessing the system. 3. Weak session handling can allow unauthorized users to stay logged in.	1. Users may be unable to access the product. 2. Unauthorized access to user accounts. 3. Loss of user trust and potential security incidents.			
2	Authorization & Dashboard Access Control	This area includes access restrictions for protected routes and visibility of user-specific data on the dashboard.	1. Dashboard contains sensitive user information. 2. Improper access control can allow direct URL access without authentication. 3. Such issues are often not visible through UI testing alone.	1. Exposure of confidential user data. 2. Privacy and compliance violations. 3. Serious impact on brand credibility.			
3	File Upload Validation & Handling	This area involves file type validation, file size checks, and backend handling of uploaded files.	1. File upload directly interacts with backend infrastructure. 2. Improper validation can allow malicious or oversized files. 3. Storage and server resources can be misused easily.	1. Server performance degradation. 2. Security vulnerabilities and malware risk. 3. Increased infrastructure and maintenance cost.			
4	Error Handling & Application Stability	This area covers how the application behaves during API failures, network issues, and unexpected system errors.	1. Backend failures are common in real environments. 2. Poor error handling can crash the UI or show blank screens. 3. Users have no recovery path in such cases.	1. Poor user experience and frustration. 2. High customer support load. 3. Increased user drop-off and churn.			
5	Logout & Session Termination	This area includes logout functionality and proper invalidation of user sessions.	1. Logout is critical for shared or public devices. 2. If session is not destroyed, users remain authenticated. 3. Such bugs are hard to notice during normal usage.	1. Privacy breaches on shared systems. 2. Unauthorized access to personal data. 3. Loss of user confidence and legal risk.			