1. **Title:** Login Functionality Test Case

ID	Test Case	Pre-condition	Test Steps	Test Data	Expected Output	Post-condition	Actual Output	Status	Comments
Login_001	Verification of Login Page with Valid Username and Password	Username and Password need to match for Log In	a. Type correct Username or Email Address b. Type correct Password c. Press on the Log In button	a. Valid Username b. Valid Password c. Button clicked	Log In Successful	WordPress Dashboard is visible	Log In Successful	Pass	
Login_002	Verification of Login Page with Valid Username and invalid Password	Username and Password need to match for Log In	a. Type correct Username or Email Address b. Type correct Password c. Press on the Log In button	a. Valid Username b. Invalid Password c. Button clicked	Error: The password you entered for the username Jacob is incorrect. Lost your password?	Same Log In page appears	As expected	Pass	
Login_003	Verification of Login Page with Invalid Username and Valid Password	Username and Password need to match for Log In	a. Type correct Username or Email Address b. Type correct Password c. Press on the Log In button	a. Invalid Username b. Valid Password c. Button clicked	Error: The username Jacob2 is not registered on this site. If you are unsure of your username, try your email address instead.	Same Log In page appears	As expected	Pass	
Login_004	Verification of Login Page with Invalid Username and Invalid Password	Username and Password need to match for Log In	a. Type correct Username or Email Address b. Type correct Password c. Press on the Log In button	a. Invalid Username b. Invalid Password c. Button clicked	Error: The username Jacob2 is not registered on this site. If you are unsure of your username, try your email address instead.	Same Log In page appears	As expected	Pass	N/A
Login_005	Verification of Login Page with Valid Email Address and Invalid Password	Username and Password need to match for Log In	a. Type correct Username or Email Address b. Type correct Password c. Press on the Log In button	a. Valid Email b. Invalid Password c. Button clicked	Error: The password you entered for the email address jacob@exceldemy.com is incorrect. Lost your password?	Same Log In page appears	As expected	Pass	
Login_006	Verification of Login Page with Valid Email Address and Valid Password	Username and Password need to match for Log In	a. Type correct Username or Email Address b. Type correct Password c. Press on the Log In button	a. Valid Email b. Valid Password c. Button clicked	Log In Successful	WordPress Dashboard is visible	Log In Successful	Pass	N/A

Login_007	Verification of Login Page with Invalid Email Address and Valid Password	Username and Password need to match for Log In	a. Type correct Username or Email Address b. Type correct Password c. Press on the Log In button	a. Invalid Email b. Valid Password c. Button clicked	Unknown email address. Check again or try your username.	Same Log In page appears	Unknown email address. Check again or try your username.	Pass	
Login_008	Verification of Login Page with Invalid Email Address and Invalid Password	Username and Password need to match for Log In	a. Type correct Username or Email Address b. Type correct Password c. Press on the Log In button	a. Invalid Email b. Invalid Password c. Button clicked	Unknown email address. Check again or try your username.	Same Log In page appears	As expected	Pass	N/A
Login_009	Verification of Login Page with Valid Username and Password	Username and Password need to match for Log In	a. Type correct Username or Email Address b. Type correct Password c. Press on the Log In button	a. Valid Username b. Valid Password c. Button clicked	Log In Successful	WordPress Dashboard is visible	Same Log In page appears	Fail	
Login_010	Verification of Login Page with Valid Email Address and Valid Password	Username and Password need to match for Log In	a. Type correct Username or Email Address b. Type correct Password c. Press on the Log In button	a. Valid Username b. Valid Password c. Button can't be clicked	Log In Successful	WordPress Dashboard is visible	Log In button not working	Fail	N/A
Login_011	Verification of ENTER key as alternative to Sign In Button	Username and Password need to match for Log In	a. Type correct Username or Email Address b. Type correct Password c. Press on ENTER button	a. Valid Username b. Valid Password c. ENTER button pressed	Log In Successful	WordPress Dashboard is visible	ENTER button not working as intended	Fail	
Login_012	Verification of Remember Me Button	Username and Password need to match for Log In	a. Type correct Username or Email Address b. Type correct Password c. Select Remember Me Checkbox d. Press on Log In Button	a. Valid Username b. Valid Password c. Log In button pressed	Log In Successful	WordPress Dashboard is visible	ENTER button not working as intended	Fail	
Login_013	Verification of CAPTCHA Button	Username and Password need to match for Log In	a. Type correct Username or Email Address b. Type correct Password c. Press on Log In Button	a. Invalid Username 3 times b. Invalid Password 3 times c. Log In button pressed 3 times	CAPTCHA shown after 2 incorrect attempts	CAPTCHA is visible	As expected	Pass	N/A
Login_014	Verification of Sign Up Button	1. Internet Browser 2. Functional Internet 3. Site Web Address Entered	1. Type website address in the web browser 2. Press Enter 3. Click on Sign Up from the Log In Page	Website address	Sign Up page is shown	Sign Up successful	As expected	Pass	

2. Unit Testing: Unit testing is a software testing technique where individual units or components of a software application are tested in isolation to ensure they perform as expected. It verifies that each unit functions correctly as per the defined specifications.

Example: In a web application, a unit test may involve testing a single function or method that calculates the total price of items in a shopping cart. The test would verify that the function returns the correct total amount based on the input items and their prices.

Smoke Testing: Smoke testing, also known as build verification testing, is an initial testing phase performed on a software build to ensure that the critical functionalities of the application work properly before proceeding with further testing. It aims to identify major issues early in the development process.

Example: After a new version of software is deployed, smoke testing involves testing essential features like login, navigation between pages, and basic functionality to confirm that the build is stable enough for further testing.

User Acceptance Testing (UAT): User Acceptance Testing (UAT) is the final phase of testing performed by end-users to validate whether the software meets their requirements and business needs. It ensures that the software is ready for release and can be used effectively in a real-world environment.

Example: In a banking application, users would perform UAT to verify that all banking transactions, account management features, and security measures meet their expectations and comply with regulatory standards.

Integration Testing: Integration Testing is a testing technique where individual units or components of a software application are combined and tested as a group. It ensures that the interactions between integrated components function correctly and that the software behaves as expected when multiple units work together.

Example*:* In an e-commerce platform, integration testing would involve testing the interaction between the shopping cart module, payment gateway, and inventory management system to verify seamless processing of orders.

Regression Testing: Regression Testing is performed to ensure that recent changes or enhancements in the software do not adversely affect the existing functionalities. It involves re-testing previously tested features to confirm that they still work correctly after modifications.

Example*:* After fixing a bug in the checkout process of an online store, regression testing would involve re-testing all other functionalities of the website to ensure that the fix did not introduce new bugs elsewhere.

Performance Testing: Performance Testing is conducted to evaluate the speed, responsiveness, and stability of a software application under various workload conditions. It measures how well the system performs in terms of scalability, reliability, and resource usage.

Example*:* Performance testing of a web application might involve simulating multiple users accessing the website simultaneously to assess its responsiveness and ability to handle peak loads without crashing or slowing down.

Load Testing: Load Testing is a type of performance testing that evaluates the behavior of a software application under expected and peak loads. It assesses the system's performance metrics, such as response time and throughput, when subjected to high user concurrency and data volumes.

Example*:* Load testing of a cloud-based file storage service would involve simulating a large number of users uploading and downloading files concurrently to determine how the system handles heavy loads and whether it maintains acceptable performance levels.

Security Testing: Security Testing is conducted to identify vulnerabilities and weaknesses in a software application's security controls and protocols. It aims to ensure that the application is protected against unauthorized access, data breaches, and other security threats.

Example: Security testing of a mobile banking app would involve testing for common security vulnerabilities such as SQL injection, cross-site scripting (XSS), and insecure authentication mechanisms to prevent potential exploits and protect sensitive user data.

Portability Testing: Portability Testing assesses the ease with which a software application can be transferred or adapted to different hardware, operating systems, or environments without requiring significant modifications. It verifies the software's compatibility and functionality across diverse platforms.

Example: Portability testing of a mobile application would involve testing its compatibility with various mobile devices (e.g., smartphones and tablets) and operating systems (e.g., iOS and Android) to ensure consistent performance and user experience across different platforms.

Accountability Testing: Accountability Testing evaluates the software's ability to accurately record and track user actions, transactions, and system activities for auditing and accountability purposes. It ensures that the software maintains accurate logs and audit trails to trace user interactions and system events.

Example*:* Accountability testing of an electronic medical records system would involve verifying that the system accurately records all patient data entries, updates, and access activities, enabling healthcare providers to track and review user actions for compliance and accountability purposes.

Reliability Testing: Reliability Testing assesses the software's ability to consistently perform its functions without failure under specified conditions for a defined period. It verifies the software's stability, robustness, and fault tolerance to ensure uninterrupted operation.

Example: Reliability testing of a web-based email service would involve subjecting the system to prolonged usage and stress testing to validate its reliability in handling high volumes of emails, attachments, and user interactions without crashing or data loss.

Efficiency Testing: Efficiency Testing evaluates the software's resource utilization, processing speed, and computational efficiency under different workloads. It aims to identify areas of optimization to improve the software's performance and minimize resource consumption.

Example: Efficiency testing of a database management system would involve measuring its query execution time, indexing efficiency, and memory usage to identify bottlenecks and optimize database performance for faster data retrieval and processing.

Volume Testing: Volume Testing evaluates the software's ability to handle large volumes of data or transactions without performance degradation or system failure. It verifies scalability and resource management under increasing data loads.

Example: Volume testing of a customer relationship management (CRM) software would involve simulating a high volume of customer records, transactions, and interactions to assess the system's responsiveness and database capacity to manage and process large datasets efficiently.

Recovery Testing: Recovery Testing assesses the software's ability to recover from system failures, errors, or disruptions and resume normal operation without data loss or corruption. It verifies the effectiveness of backup and recovery mechanisms.

Example: Recovery testing of an online banking system would involve intentionally disrupting the system, such as simulating server crashes or network failures, and verifying that the system can recover gracefully and restore user data and transactions to their pre-failure state.

Responsive Testing: Responsive Testing evaluates the software's responsiveness and user interface behavior across different devices, screen sizes, and orientations. It ensures that the application displays and functions correctly on various platforms and devices.

Example: Responsive testing of a responsive web application would involve testing its layout, navigation, and interactive elements on desktops, laptops, tablets, and smartphones to ensure consistent user experience and functionality across different screen sizes and devices.

Visual Testing: Visual Testing verifies the visual appearance and layout of a software application's user interface components, such as buttons, icons, images, and text, across different browsers, devices, and resolutions. It ensures that the application's visual elements are displayed correctly and consistently.

Example: Visual testing of a web application would involve comparing screenshots of each page or screen across different browsers (e.g., Chrome, Firefox, Safari) and devices (e.g., desktop, tablet, mobile) to identify any visual discrepancies or rendering issues that may affect the user experience.