Banking Application Test Case Templates	Pre Conditions	Expected Result	Actual Result	Post Condition	Pass/Fail	Test Owner
UI Functionality Testing for Banking Application						
Login, Security, and Authentication Process: Ensure login functionality works properly with legitimate credentials. Test multiple authentication factors before login. Display error messages for users with invalid credentials or disabilities.						
Application Responsiveness: Ensure the application loads quickly and displays appropriate results when reloaded by the user. The application should be responsive and supported on all screen sizes, desktops, and tablets, regardless of the operating system. Note: Test your Website responsiveness on LambdaTest Real device cloud.						
Compatibility with Different Browsers: Ensure the application is compatible with popular browsers such as Firefox, Chrome, Edge, Internet Explorer, and Opera.						
Navigation: Ensure smooth user navigation through menus, buttons (home, exit, login/logout), and links. Ensure that menus and buttons direct users to the intended pages and are properly aligned and understandable. Ensure compatibility with different screen sizes.						
Account Management: Verify the user can create, delete, link, update, and manage accounts. Ensure changes are reflected in the user account information appropriately and promptly.						
Error Handling and Error Messages: Ensure meaningful error messages are displayed during session timeouts, connectivity issues, server errors, or data validation errors. Ensure the application recovers quickly from these errors.						
Alerts and Notifications: Ensure users receive alerts for important updates like low balance, credit/debit activities, and login activities on other devices. Ensure the timing and accuracy of notifications.						
Input Fields: Ensure input fields are accessible and of the correct data types. Ensure error messages are displayed for invalid entries, allowing the user to re-enter data. Test input field limits for functionality.						
Accessibility: Ensure the application is user-friendly for people with disabilities (e.g., visual impairment). Verify color contrast, keyboard navigation, screen reader compatibility, and multi-language support. Ensure voice integration is enabled for accessibility.						
Feedback, FAQs, Help Desk: Ensure users can submit feedback and complaints, receiving prompt replies. Ensure FAQs are up-to-date and helpful. Ensure 24/7 access to a help/support panel.						
Transactions-related testing						
Internal Account Transfers Ensure multiple accounts of the user are properly linked to the application. Ensure funds can be transferred successfully between the internal accounts of the user.						
External Account Transfers Ensure the user account can be successfully linked to external accounts once the necessary details are entered. Ensure the transfer completes successfully to the external account.						

Non-Existing Account Transfers Ensure that attempts to transfer funds to a non-existing account are prohibited and the user is notified.						
Transaction Success/Failure Ensure that the user is immediately notified of the success or failure of a transaction.						
Transaction Reference Number Ensure that a unique transaction reference number is generated for each transaction. Ensure the reference number is delivered to the user.						
Insufficient Balance Ensure that when a transfer exceeds the available balance in the user account, the transfer is restricted and the user is informed.						
Transaction History Ensure that the transactions performed by the user are properly maintained and accessible at any time.						
Transfer Limit Ensure that transactions are not allowed beyond a predefined limit. Ensure the user is notified when the maximum transaction limit is reached.						
Testing related to online payments						
Payment Gateway Integration: Ensure the payment gateway is properly integrated with the application for transferring funds to e-commerce websites.						
Initiations Ensure that the user can successfully initiate payments.						
Payment Method Ensure the user can easily choose from available payment methods, including credit cards, internet banking, debit cards, UPIs, Paytm, and GPay.						
Payee Selection Ensure the payer can choose the payee through simple search options effectively.						
Invalid Payee Ensure the user is notified if invalid payee details are entered, with appropriate error messages when the intended payee doesn't have a net banking account.						
Invalid Details Ensure the user is properly notified about invalid or mismatched bill details provided during the payment process.						
Scheduled/Recurring Payments Ensure that users can set up recurring or scheduled payments. Ensure the application triggers the payment at the scheduled time and sends notifications to the user before and after the payment.						
Confirmation Ensure the user receives a notification confirming the bill payment, including all relevant payment details.						

Payment History Ensure the application keeps a record of all past payments made by the user, which is accessible at any time.						
Payment Amount Limit Ensure transactions are only processed within the allowed limits. Notify the user if the entered payment amount exceeds the set limit. Ensure that any transaction after a specified number of attempts is prohibited.						
Payment Failure Ensure the user is notified in case of payment failure, with the reason for failure clearly provided.						
Testing related to bill payments						
Bill Addition Ensure that a new biller can be added at any time for online bill payments.						
Bill Payment Validation Ensure the validity of the biller before processing any payment to avoid transactions with non-existing or invalid billers.						
Cancellation Ensure the user can cancel any scheduled payments to the biller at any time. Ensure a confirmation message is displayed once the cancellation is successful.						
Successful Bill Payment Ensure the user receives a notification message after a successful bill payment.						
Bill Management Ensure the user can view or download the bills after a successful payment whenever needed.						
Bill Payment Failure Ensure the user is notified in case of bill payment failure, with the reason for failure clearly provided.						
Bill Payment History Ensure that the history of all bill payments is maintained accurately and is accessible to the user at any time.						
Testing related to account updation						
Account Preferences Ensure the user can switch between multiple accounts after a verified login.						
Beneficiary Updation Ensure the user can add or remove account beneficiaries whenever needed.						
Account Information Ensure the user can update their account information, including personal details, contact information, security details, and beneficiary details.						
Account Information Incompletion Ensure the user is notified if any essential information is left incomplete during the account update process.						

Successful Updation Ensure that after the user updates their profile, the updated information is properly retrieved by the application and reflected in the user's account.						
Privacy and Security Ensure that user details are secured throughout the account updation process.						
Login Ensure the user can successfully log in after the account information has been updated.						
Testing related to debit or credit cards						
Card Enrollment Ensure the user can register their credit or debit cards successfully. Ensure the user receives a notification confirming successful card registration.						
Card Validation Ensure the application validates card details, including the card expiry date, PIN, etc., during registration or updation.						
Blocking Cards Ensure the user can block their credit/debit cards whenever needed. Ensure that no future transactions can be made on the blocked card.						
Unblocking Cards Ensure the user can unblock their credit/debit cards whenever needed. Ensure that future transactions are allowed once the card is unblocked.						
Card Details Updation Ensure the user can update their card details (e.g., PIN number) on the application. Ensure the updated details reflect properly on future transactions.						
Card Requisition Ensure the user can request a new credit/debit card successfully. Ensure the user is notified about the card replacement and its delivery status.						
Card Limits Ensure the credit card limits are correctly displayed to the user.						
Credit Limit Increase/Decrease Ensure the user can request an increase or decrease in their credit limits. Ensure that the approved changes are reflected properly in the application, regardless of past transactions.						
Usage Notifications Ensure the user receives timely notifications when they are about to reach or have reached their maximum credit limit.						
Multiple Account Credit Limit Ensure that credit limits are maintained properly for all accounts linked to the user's profile.						

Credit Limit History Ensure the user can view the history of credit limit changes.						
Temporary Suspension Ensure the credit card limit is temporarily suspended in case of suspicious activity. Ensure the user is informed about the suspension and that it is lifted once the issue is resolved.						
Testing related to disputes on transactions						
Requisites Ensure that the user is prompted with all necessary details to submit a dispute request when they opt to raise a dispute.						
Initiation Ensure the user can raise a dispute on the desired transaction by selecting the appropriate category. Ensure that the dispute request is submitted successfully.						
Dispute Tracking Ensure the user can track and monitor the progress of their dispute request successfully.						
Dispute Resolution Ensure the application provides options for resolving the dispute, such as transaction reversal, refunding, negotiation, or credit adjustment, appropriately.						
Dispute Intensification Ensure the user can escalate their dispute and request involvement from a superior successfully.						
Parallel Disputes Management Ensure the application can manage multiple disputes raised by the user simultaneously. Ensure related messages are sent to the user for each dispute.						
Finalized Dispute Result Ensure the user is notified of the finalized decision on their dispute, whether it's a rejection or acceptance, or any other decision.						
Closure Ensure that the dispute status is updated after the decision is made and successfully notified to the user.						
Report Generation Ensure the application can generate a detailed report of the entire dispute resolution process to improve efficiency.						
Dispute Investigation Ensure that the dispute investigation process is able to successfully review the transaction history, available evidence, and customer statement.						
Testing related to loan applications and payment methods						
Eligibility Verification Ensure only eligible applicants can apply for a loan based on factors like credit history, CIBIL score, income, employment status, and other relevant criteria.						

Pre-requisites Ensure the application prompts the user to upload all necessary documents, such as identification documents, salary slips, proof of address, and bank statements for the last three months.						
Validation Ensure the application validates the applicant's details during entry and provides error messages for invalid fields.						
Loan Amount Calculation Ensure the application successfully calculates the loan amount sanctioned to the customer using the provided information.						
Interest Calculation Ensure the application accurately calculates the interest amount and interest rate based on the loan amount and tenure.						
Successful Submission Ensure the loan application is submitted successfully after the applicant uploads all essential documents.						
Status Tracking Ensure the user can track the progress of their loan application and receive timely updates and notifications about the application status.						
Application Approval Ensure the application can successfully approve eligible loan applications and notify the user about the approval via email or message.						
Application Disapproval Ensure the application can successfully reject ineligible loan applications and notify the applicant with appropriate reasons in a timely manner.						
Fund Transfer/Disbursement Ensure the loan amount is transferred to the designated customer's account accurately once the loan is approved.						
Loan Deferment/Forbearance Ensure the user can request a temporary postponement of loan payments and is notified about the process and final decision on time.						
Loan Payment Methods Ensure the user can pay loan amounts through any payment method, including Google Pay, Paytm, and others.						
Loan Cancellation Ensure the loan cancellation process initiated by the user is successfully processed by the application, and progress and confirmation notifications are sent to the user.						
Loan Modification Ensure the application can handle modifications requested by the user regarding the loan amount, interest rates, or tenure.						

Scheduled Payment Ensure the user can schedule, reschedule, or cancel any scheduled loan payments as needed.						
Testing related to investment accounts						
Prerequisites Ensure the application prompts the user to upload all necessary documents such as PAN card, passport, voter ID card, and other relevant documents.						
Account Creation Ensure the user can successfully create an investment account. Ensure the user receives a confirmation once the account is created. Ensure the user is notified if any field is left incomplete or invalid details are entered, with appropriate error messages.						
Investment Options and Risks Ensure the application provides the user with all available investment account types. Ensure the user is given complete information about risks, funding options, duration, and other relevant matters related to the investment.						
Funding Ensure the application can successfully manage both small and large fundings. Ensure the user can fund multiple accounts through any payment option on any device.						
Account Updation Ensure the application calculates the balance correctly after every transaction or investment. Ensure the updated balance reflects accurately after every change.						
History and Statement Reports Ensure the user can track transaction history accurately. Ensure the application can generate transaction statements and reports that include all relevant details. Ensure the user can download these reports and statements in any file format.						
Notifications Ensure the user is notified about any changes in their investment portfolio promptly.						
Account Closure Ensure the investment account can be closed successfully. Ensure that any transactions to and from the closed account are blocked. Ensure the account is cleared and handled properly after closure.						
Testing related to stock management						
Prerequisites Ensure the application prompts the user to upload all required documents such as proof of identity, proof of income, proof of bank account, PAN card, and other relevant documents before creating a stock market account.						
Account Creation Ensure the application can successfully create a stock market account after the user submits all necessary documents. Ensure the user receives a confirmation message once the account is created.						

Nature of the Account Ensure the stock market account starts with zero balance and can handle both small and large funds. Ensure the account does not run on negative balance and that updates are accurately reflected.						
Stock Search Ensure the user can search for stock details by entering the ticker symbol. Ensure the application handles cases where the stock details are unavailable in the database and displays an appropriate message.						
Current Updates Ensure the user receives regular updates on real-time stock prices, trade prices, and other relevant information. Ensure the application notifies the user about changes in the stock market in a timely manner.						
Portfolio Update Ensure the portfolio updates accurately when stocks are added or removed from the account. Ensure changes in the portfolio are saved and reflected correctly.						
Stock Purchase Ensure the user can successfully purchase stock by entering appropriate details (e.g., price, quantity, ticker symbol, type of trade). Ensure the user is notified of the successful purchase, and the portfolio balance is updated accordingly.						
Stock Selling Ensure the user can successfully sell stocks after entering valid details (e.g., shares, price, ticker symbol). Ensure appropriate error messages are displayed if invalid details are entered. Ensure the stock is removed from the portfolio once sold.						
Notifications Ensure the user is notified about stock market changes, stock purchases and sales, price changes, and custom alerts. Ensure the user can customize alerts, change frequencies, and make adjustments in the alert settings.						
Insufficient Quantity Ensure the user is notified with an error message when attempting to sell a quantity that is not available in their account.						
Diversity in Stock Selling Ensure the application handles partial stock selling (e.g., selling half the stock) correctly. Ensure the remaining stock is kept in the portfolio, and the sold stock is removed.						
Invalid Detail Entry Ensure the user is notified with an error message when invalid details are entered during stock purchase or sale (e.g., negative values, zero, out-of-market price, or incorrect ticker symbol).						
Account Statement Ensure the user can access, view, and print their account statement, which includes balance and transaction details, in formats such as .pdf, .csv, or .png.						
Trading Hours Ensure the application displays an appropriate error message if the user attempts a transaction outside of active trading hours.						

Multiple User Activities Ensure the application can handle multiple user activities simultaneously, such as one user selling a stock and another buying a stock, without errors or delays.						
Market and Limit Orders Ensure the application works as expected for both market and limit orders.						
Transaction History Ensure the user can access their transaction history, which includes sold stocks, purchased stocks, withdrawals, and other related details. Ensure transaction history is updated continuously and is properly tracked.						
Account Closure Ensure the account can be closed successfully. Ensure no further actions can be taken in the stock market using the closed account.						
Testing related to retirement accounts						
Prerequisites Ensure the application prompts the user to upload all necessary documents, such as proof of identity, pension paperwork, PAN card, and other relevant documents to create a retirement account.						
Account Creation Ensure the application can successfully create a retirement account after the user submits all required documents. Ensure the user receives an appropriate confirmation message once the account is created.						
Account User Type Ensure the user can select the desired account type, including options such as Simple IRA, Mutual Fund, Traditional IRA, Defined Contribution Plan, etc.						
Investment and Withdrawals Ensure the user can successfully contribute to or withdraw from their retirement account. Ensure confirmation messages are sent after contributions or withdrawals. Ensure the application can handle different types of withdrawals, such as partial, early, and half withdrawals.						
Changes Handling Ensure the application can handle changes to the user's investment plan midway. Ensure that any changes to the plan are reflected appropriately in the user’s account.						
Account Balance Updates Ensure the account balance is updated successfully after every transaction, including withdrawals, contributions, and rollovers. Ensure the updated balance is reflected in future transactions.						
Account Statements Ensure the user can view their account statement, which includes transaction history, dates, amounts, and descriptions. Ensure the user can download and print the statement in any format, such as .csv or .pdf.						

Notifications Ensure the user receives notifications or alerts regarding policy updates, contribution limits, or other important updates related to their retirement account.						
Account Management Ensure the user can successfully update their account with personal details, contact information, and security information. Ensure the application provides an appropriate error message when invalid details are entered. Ensure the changes reflect in the account settings and in future transactions.						
Linking Ensure the application supports integration with external systems for setting up automatic contributions. Ensure the application can validate interactions between retirement accounts and other banking features.						
Account Closure Ensure the user can close their retirement account successfully. Ensure that once the account is closed, no further transactions can be made, and the account is handled appropriately.						
Testing related to online account creation						
Prerequisites Ensure the user is prompted to upload all necessary documents, such as proof of identity, address proofs, and other relevant documents to create an online account.						
Document and Details Validation Ensure the application validates the information provided by the user. Ensure documents are stored properly in the database. Ensure any invalid or incomplete data entries are flagged and an appropriate error message is shown to the user.						
Verification Ensure the details entered by the user undergo verification. Ensure the verification process includes signature matching and facial recognition. Ensure the application can verify all types of addresses (indigenous and international), including special cases and non-standard formats. Ensure appropriate error messages are displayed when address verification fails.						
Account Creation Ensure the application can successfully create an account online for the user after submitting all necessary documents. Ensure the user receives an appropriate confirmation message once the account is created.						
Account User Type Ensure the user can create an account of the desired type. Ensure the account type options include savings account, fixed deposit account, recurring deposit account, NRI accounts, etc.						
Configuration Ensure the application properly applies configurations, additional features, and customizations made by the user. Ensure default values are automatically set by the application when the user does not provide them.						

Funding Ensure the user can successfully fund their account via online transfers, check deposits, or UPI payments. Ensure the account balance is updated appropriately after each transaction.						
Transactions Ensure transactions on the newly created account update the balance correctly. Ensure the updated balance is reflected in future transactions.						
Notifications and Alerts Ensure the application notifies the user of any changes in the bank's policies related to accounts. Ensure the user receives appropriate confirmation messages after every transaction. Ensure the user is notified of any illegitimate login activities with warning messages.						
Duplicate Accounts Creation Ensure the application prevents the user from creating multiple accounts with the same information. Ensure the application throws an appropriate warning when an attempt is made to create a duplicate account.						
Integration Ensure the application integrates with other banking systems and third-party services for secure online transactions. Ensure transparency in how the user's data is handled across services.						
History and Statement Reports Ensure the user can track their transaction history accurately. Ensure the application can generate transaction statements and reports with all the relevant details. Ensure the user can download the reports and statements in various file formats (e.g., .pdf, .csv) as needed.						
Account Closure Ensure the user can close their account successfully. Ensure the account is properly closed, and no further transactions can be made on the account.						
Testing related to online customer support						
Connecting Ensure that the user can successfully connect to the online customer support system after a legitimate login.						
Technical Support Ensure customer support can assist users with technical issues, such as application installation problems, login issues, etc. Ensure relevant details (e.g., operating system, device type, model version) are collected to assist in troubleshooting. Ensure customer support provides appropriate troubleshooting steps.						
Login Issues Ensure the application assists users with login issues by providing the right solutions based on the severity of the problem. Ensure customer support follows up with the user until the issue is resolved.						
Transaction Disputes Ensure customer support can address and resolve any queries related to transaction disputes. Ensure relevant dates and processes related to the dispute are communicated to the user.						

Queries Related to Account Balance Ensure customer support can provide accurate information regarding account balances, transactions, and account statements whenever the customer inquires.						
Problem-Related to Cards Ensure customer support can handle inquiries related to credit or debit cards, including lost or stolen card reports. Ensure the customer is guided through the process of new card requisition or replacement						
Availability Ensure online customer support is available 24/7 and responds promptly to user queries. Ensure no delays in providing troubleshooting steps and resolving issues.						
Testing related to online support tickets						
Creation Ensure the customer can create an online support ticket for assistance and issue resolution.						
Submission Ensure that when a customer creates a support ticket, a unique ID is generated for the ticket. Ensure the customer receives a confirmation message about the successful creation of the support ticket.						
Categorization Ensure the support ticket is automatically categorized based on the nature of the issue.						
Ticket Allocation Ensure the application allocates tickets to the appropriate teams or support agents based on the issue type and required expertise.						
Prioritization Ensure the application prioritizes support tickets based on the issue's impact and urgency. Ensure high-priority tickets receive instant acknowledgment and quicker resolutions, while lower-priority tickets are still addressed within an acceptable timeframe.						
Tracking Ensure the customer can track the status of their support ticket by entering the ticket ID. Ensure the application displays an appropriate message such as "ticket submitted" or "ticket resolved" as the ticket progresses.						
Escalation Ensure the customer can escalate the ticket to a higher official or elevate the ticket's priority when needed. Ensure the escalated ticket receives maximum attention from the support team.						
Ticket Closure Ensure the status of the ticket is updated once the issue is resolved and the customer is notified. Ensure the ticket is closed appropriately after resolution.						
Testing related to reward points						

Eligibility for Reward Points and Their Calculation Ensure reward points are awarded only for eligible transactions (e.g., withdrawals, deposits, transfers). Ensure the correct reward points are added to the user's account based on the transaction amount. Ensure the reward points calculation reflects appropriately in future transactions.						
Display Ensure the application displays the appropriate reward points information. Ensure any updates in reward points after a transaction or withdrawal reflect immediately in the user's account. Ensure the transaction history related to reward points (e.g., earned points, redeemed points) is displayed accurately by the application.						
Redemption Ensure the user has sufficient reward points for any redemption request. Ensure the redemption process is correctly handled and the redeemed points are deducted from the user's account. Ensure the user fully benefits from the redeemed points as expected.						
Expiration Ensure that expired reward points are deducted from the user's account. Ensure that valid reward points remain in the user's account while expired points are removed.						
Testing related to credit score and credit report						
Estimation Ensure the application calculates the credit score based on the user's credit history. Ensure the estimated credit score matches the expected value based on the user's credit history.						
Threshold Ensure that any action or service (e.g., loan approval, credit card offers) is only offered to the user if their credit score is above the required threshold value.						
Accuracy Ensure the credit report generated is accurate and reflects the correct information. Ensure any modifications or changes to the credit history are updated in the credit report appropriately. Ensure the credit report is viewable, manageable, and easily accessible in any format (e.g., PDF, CSV) to the user.						
Retrieval Ensure the application can retrieve and display the credit report to the user. Ensure the credit report includes all relevant details such as credit history, payment history, credit limit, etc.						
Dispute Clearance Ensure the user can raise disputes for any incorrect information on their credit report. Ensure the disputes are processed appropriately for resolution.						
Testing related to overdraft protection						
Access, Enable, and Disable Ensure the user can access the overdraft protection feature in their account settings. Ensure the user can turn the overdraft protection feature on or off whenever needed. Ensure the user is notified about the update, and the changes are reflected in future bank-related actions.						

Threshold Setting Ensure the user can set a desired threshold or limit for their overdraft protection feature. Ensure the threshold is appropriately saved and applied to the account.						
Triggers and Alerts Ensure the overdraft protection is triggered when the account balance falls below the set threshold. Ensure the user receives timely notifications and alerts about the overdraft protection being activated.						
Payment Ensure the user can repay their overdraft balance. Ensure the negative or zero balance is converted to a positive balance once the repayment is processed.						
Fee Calculation Ensure the application calculates the overdraft fee appropriately when the balance falls below the set threshold. Ensure the fee is correctly applied and reflected in the user's account.						
Testing related to safe deposit box						
Vacancy Ensure the user can check the availability of safe deposit boxes through the application. Ensure the application displays the status of the deposit box (e.g., booked, reserved).						
Reservation and Accessibility Ensure the user can reserve a safe deposit box by specifying the size and location of the box. Ensure the reservation is confirmed through a notification or confirmation message. Ensure the booked safe deposit box is unavailable for others until it's freed. Ensure the reserved box is accessible and manageable by the user at any time.						
Management Ensure the application allows the user to update information related to the safe deposit box (e.g., change size, location). Ensure the user can extend the lease, cancel the reservation, or raise disputes. Ensure the user can view, add, or remove items from the safe deposit box. Ensure all changes are successfully updated in the system.						
Testing related to annuity accounts						
Information Input Ensure all fields required to open an annuity account are displayed when the user accesses the annuity section. Ensure the application notifies the user of any invalid data entry or incomplete fields with appropriate error messages. Ensure default values are automatically filled in the fields. Ensure the submission is successful once the details are filled and proceed to the next step.						
Account Type Ensure the user can select the type of annuity account they want (e.g., immediate, deferred, fixed, variable). Ensure the selected account type is saved and updated correctly.						
Beneficiary Details Ensure the user is provided with an option to designate a beneficiary for their annuity account. Ensure the beneficiary details are stored, updated, and properly linked to the annuity account.						

Contribution Amount Setup Ensure the user can specify the contribution amount for their annuity account. Ensure the application prevents the user from entering negative or zero values for the contribution amount, displaying an appropriate error message when necessary.						
Contribution Ensure the user can set up automatic contributions to their annuity account. Ensure contributions are successfully triggered on time. Ensure notifications are sent to the user before and after the contribution.						
Submission Ensure the applicant can successfully submit their annuity account application. Ensure an appropriate confirmation message is displayed after successful submission.						
Account Closure Ensure the user can successfully close their annuity account when requested. Ensure no further transactions are allowed on the closed account, and it is properly handled by the application.						
Testing related to estate planning documents						
Information Input Ensure that all required fields for the estate planning documents are displayed when the user accesses the annuity section. Ensure the application notifies the user of any invalid data entry or incomplete fields, providing an appropriate error message. Ensure default values are automatically filled where applicable. Ensure successful submission once the details are provided, and the page proceeds to the next step.						
Account Type Ensure the user can select the desired annuity account type (e.g., immediate, deferred, fixed, variable). Ensure the selected account type is stored and updated appropriately in the system.						
Beneficiary Details Ensure the user can designate a beneficiary for their estate planning documents. Ensure the beneficiary details are stored and linked to the annuity account, and the information is updated successfully.						
Contribution Amount Setup Ensure the user can specify the contribution amount for their annuity account. Ensure the system prevents negative or zero values in the contribution field, displaying an appropriate error message when necessary.						
Contribution Ensure the user can set up automatic contributions to their annuity account. Ensure contributions are successfully triggered on time, and the user receives notifications before and after the contributions.						
Submission Ensure the user can successfully submit their annuity account application for estate planning. Ensure the user receives an appropriate confirmation message after submission.						

Account Closure Ensure the user can request to close their annuity account. Ensure that the account is closed appropriately, with no further transactions allowed, and the process is properly handled.						
Testing related to online security						
Security Management Ensure the application guides the user to set strong passwords that include a combination of letters, special characters, and numbers. Ensure the authentication mechanism is highly secure, prohibiting any illegitimate login attempts (e.g., brute force, credential stuffing).						
Encryption Ensure that sensitive user data (e.g., passwords, personal information) is encrypted in the database. Ensure the data is secured from unauthorized access, brute-force attacks, or data phishing.						
Password Reset Ensure the "Forgot Password" option is available for users who lose track of their password. Ensure the password reset process is secure, with a time-sensitive reset link sent to the user’s email. Ensure the reset link expires after a short time once successfully used.						
Session Management Ensure the application generates a secure session identifier for each user session. Ensure the session identifier is transmitted securely and stored in a way that prevents unauthorized access. Ensure sessions expire after a defined period of inactivity to prevent hijacking or session fixation.						
Error and Vulnerability Handling Ensure the application handles errors gracefully without revealing sensitive information. Ensure that security vulnerabilities (e.g., broken access control, insecure design, security misconfigurations) are handled appropriately without exposing user data or system details.						
Testing related to logout						
Confirmation Prompt Ensure that when the user chooses the log-out option, a confirmation prompt is displayed asking them to confirm the logout. Ensure the prompt includes a "Cancel Logout" option, allowing the user to opt-out of logging out if they change their mind.						
Successful Logout Ensure that the user can successfully log out when they click the "Logout" option. Ensure a confirmation message is displayed after successful logout or the user is redirected to the login page.						
Handling Invalid Logout Ensure that if an invalid logout attempt is made (e.g., due to session issues), the application denies the logout and displays an appropriate error message. Ensure the user is redirected to an error page or provided with further instructions to resolve the issue.						

Session Timeout Ensure that after a period of inactivity, the application automatically logs the user out. Ensure the user is either redirected to the login page or shown a session-expired message upon successful logout due to inactivity.						