TEST CASE SPECIFICATION

FinTrack — Target Banking Application

BugVault SQA Hackathon

Version	1.0
Format	Gherkin / BDD (Given-When-Then)
Coverage	Functional, UI/UX, Boundary, Negative
Bug Flags	Included — Tier 1 & Tier 2 marked
Total Modules	6
Total Test Cases	60+
Prepared For	Development Team

How to Read This Document

This document contains Gherkin-style (Given-When-Then) test cases for every functional area of FinTrack, the target banking application used in the BugVault SQA Hackathon. It serves two purposes:

- Development Guide: Developers can use each test case to understand exactly what behavior to implement in the application.
- AI Prompt Reference: Each test case block can be fed directly to an AI code generator (e.g., Claude Code) as precise, structured behavior specifications.

Legend

@functional	Standard functional behavior test — implement exactly as described.
@ui	UI/UX behavior test — governs visual, layout, or interaction behavior.
@boundary	Boundary/edge case — test the limits of valid input ranges.
@negative	Negative test — system must reject or handle invalid input gracefully.
@bug_t1	TIER 1 BUG FLAG — developer must intentionally break this behavior (10 pts bug).
@bug_t2	TIER 2 BUG FLAG — developer must intentionally break this behavior in a hidden way (50 pts bug).

Bug-flagged test cases have their Gherkin block highlighted in red. The 'INTENTIONAL BUG FLAG' banner below the test case tells the developer exactly what incorrect behavior to code in place of the correct behavior described in the 'Then' steps.

Non-flagged test cases must be implemented correctly — they represent the application's legitimate, expected behavior.

🔐 MOD-01 — User Authentication

TC-AUTH-01 — Successful Login

Test Case ID	TC-AUTH-01
Module	Authentication
Type	Functional
Priority	Critical

```
@functional
Feature: User Authentication

  Scenario: Participant logs in with valid credentials
    Given the participant is on the FinTrack login page
    And the participant has been provided valid credentials (email and password)
    When the participant enters their registered email address
    And the participant enters their correct password
    And the participant clicks the 'Login' button
    Then the system authenticates the participant
    And the participant is redirected to the Dashboard page
    And the dashboard displays a welcome message with the participant's name
    And the session is marked as active
```

✅ PASS Condition
Participant lands on Dashboard with their name displayed and session active.

❌ FAIL Condition
Login fails, error shown, or wrong page is loaded.

TC-AUTH-02 — Login with Wrong Password

Test Case ID	TC-AUTH-02
Module	Authentication
Type	Negative
Priority	Critical

```
@negative
Feature: User Authentication

  Scenario: Participant enters an incorrect password
```

```
    Given the participant is on the FinTrack login page
    When the participant enters their registered email address
    And the participant enters an incorrect password
    And clicks the 'Login' button
    Then the system rejects the login attempt
    And a generic error message is displayed: 'Invalid email or password'
    And the password field is cleared
    And the participant remains on the login page
    And no session is created
```

✅ **PASS Condition**	❌ **FAIL Condition**
Error message shown. No session created. Participant stays on login page.	Login succeeds with wrong password, or error message reveals which field is wrong.

TC-AUTH-03 — Login with Empty Fields

Test Case ID	TC-AUTH-03
Module	Authentication
Type	Negative
Priority	High

```
@negative
Feature: User Authentication

  Scenario: Participant submits the login form with empty fields
    Given the participant is on the FinTrack login page
    When the participant leaves the email field empty
    And the participant leaves the password field empty
    And the participant clicks the 'Login' button
    Then inline validation errors appear below each empty field
    And the form is NOT submitted
    And no API call is made to the authentication endpoint
```

✅ **PASS Condition**	❌ **FAIL Condition**
Inline validation errors appear. Form not submitted.	Form submits with empty fields or shows a generic page error.

TC-AUTH-04 — Forgot Password Link

Test Case ID	TC-AUTH-04
Module	Authentication

Type	Functional | Bug Flag
Priority	Medium

```
@functional @bug_t1
Feature: User Authentication

  Scenario: Participant clicks the Forgot Password link
     Given the participant is on the FinTrack login page
     And a 'Forgot Password' link is visible below the password field
     When the participant clicks the 'Forgot Password' link
     Then the participant should be navigated to a password recovery page
     # [CORRECT behavior — but this must be broken as per bug flag below]
```

✅ **PASS Condition** [SHOULD BE] Participant navigates to a password recovery page.	❌ **FAIL Condition** [ACTUAL — BUG] Participant hits a 404 page or dead-end URL.

🐛 [INTENTIONAL BUG FLAG] T1-001 | TIER 1 — COMMON BUG (10 pts)

Plant bug: Make the 'Forgot Password' link navigate to a non-existent route (e.g., /password-reset) that returns a 404 page. The link must be visually present and clickable.

TC-AUTH-05 — Session Persistence

Test Case ID	TC-AUTH-05
Module	Authentication
Type	Functional
Priority	High

```
@functional
Feature: User Authentication

  Scenario: Session remains active after browser refresh
     Given the participant is logged into FinTrack
     When the participant refreshes the browser page
     Then the participant remains logged in
     And is NOT redirected to the login page
     And all dashboard data reloads correctly
```

✅ **PASS Condition** Participant stays logged in after refresh. Dashboard loads normally.	❌ **FAIL Condition** Participant is logged out on refresh and redirected to login.

TC-AUTH-06 — Logout

Test Case ID	TC-AUTH-06
Module	Authentication
Type	Functional
Priority	High

```
@functional
Feature: User Authentication

  Scenario: Participant logs out successfully
    Given the participant is logged into FinTrack
    When the participant clicks the 'Logout' button or menu item
    Then the session is terminated server-side
    And the participant is redirected to the login page
    And attempting to navigate back via browser back button does not restore the session
    And any protected route redirects to login
```

✅ **PASS Condition**	❌ **FAIL Condition**
Session ended. Redirected to login. Back button does not restore session.	Session persists after logout or back button restores access.

🏠 MOD-02 — Dashboard

TC-DASH-01 — Dashboard Data Display

Test Case ID	TC-DASH-01
Module	Dashboard
Type	Functional
Priority	Critical

```
@functional
Feature: Dashboard

  Scenario: Participant views the dashboard after login
    Given the participant is logged into FinTrack
    When the participant is on the Dashboard page
    Then the following elements are visible and correctly populated:
    And account holder full name is displayed
    And account number is displayed (partially masked: first 4 and last 4
digits visible)
    And current account balance is displayed in BDT with 2 decimal places
    And account type label (Savings / Current) is displayed
    And a list of the 5 most recent transactions is shown
    And quick action buttons are visible: 'Transfer Funds', 'Pay Bills',
'View Statement'
```

✅ **PASS Condition**
All data elements load correctly and are populated with real account data.

❌ **FAIL Condition**
Any element is missing, empty, or shows placeholder text like 'N/A' or '0.00' for a funded account.

TC-DASH-02 — Spending Chart Axis Labels

Test Case ID	TC-DASH-02
Module	Dashboard
Type	UI/UX | Bug Flag
Priority	Medium

```
@ui @bug_t1
Feature: Dashboard
```

```
 Scenario: Spending chart displays correct axis labels
    Given the participant is on the Dashboard page
    When the spending/transaction chart is rendered
    Then the Y-axis should be labeled 'BDT' or 'Amount (BDT)'
    And the X-axis should show date labels (days or months)
    And the chart title should read 'Transaction Overview — Last 30 Days'
    # [CORRECT behavior — but Y-axis label must be broken as per bug flag]
```

✅ PASS Condition
[SHOULD BE] Y-axis label reads 'BDT' or 'Amount (BDT)'.

❌ FAIL Condition
[ACTUAL — BUG] Y-axis label reads 'Units' — an incorrect/meaningless label.

🐛 [INTENTIONAL BUG FLAG] T1-004 | TIER 1 — COMMON BUG (10 pts)
Plant bug: Hardcode the Y-axis label of the dashboard chart to 'Units' instead of 'BDT' or 'Amount (BDT)'. Everything else on the chart must render correctly.

TC-DASH-03 — Recent Transactions Update After Transfer

Test Case ID	TC-DASH-03
Module	Dashboard
Type	Functional | Bug Flag
Priority	High

```
@functional @bug_t1
Feature: Dashboard

 Scenario: Recent transactions reflect latest completed transfer
    Given the participant is logged into FinTrack
    And the participant completes a fund transfer successfully
    When the participant navigates back to the Dashboard
    Then the 'Recent Transactions' list should update automatically
    And the new transfer should appear as the most recent entry
    And the displayed balance should reflect the deducted amount
    # [CORRECT behavior — but must be broken as per bug flag]
```

✅ PASS Condition
[SHOULD BE] Recent transactions and balance update immediately on Dashboard return.

❌ FAIL Condition
[ACTUAL — BUG] Dashboard still shows old transaction list and old balance until a hard page reload (F5).

🐛 [INTENTIONAL BUG FLAG] T1-009 | TIER 1 — COMMON BUG (10 pts)

Plant bug: Do NOT refresh the dashboard data on route navigation — only refresh on a full browser reload (F5/Ctrl+R). The dashboard should cache the old state when navigated back to.

TC-DASH-04 — Phantom Balance at Exactly BDT 10,000

Test Case ID	TC-DASH-04
Module	Dashboard
Type	Boundary | Bug Flag
Priority	Critical

```
@boundary @bug_t2
Feature: Dashboard

  Scenario: Balance displays accurately when account reaches exactly BDT 10,000
     Given the participant's account balance is near BDT 10,000
     When a transaction brings the balance to exactly BDT 10,000.00
     Then the dashboard should display the balance as BDT 10,000.00
     And NO bonus or adjustment should be added to the displayed balance
     And the displayed balance must match the actual ledger balance
     # [CORRECT behavior — must be broken as per the Tier 2 bug flag below]
```

✅ **PASS Condition**	❌ **FAIL Condition**
[SHOULD BE] Balance shows exactly BDT 10,000.00. No bonus applied. UI matches ledger.	[ACTUAL — BUG] UI shows BDT 10,500.00 (phantom +500 loyalty bonus added to display only). Ledger still holds BDT 10,000.00 — permanent mismatch.

🐛 **[INTENTIONAL BUG FLAG] T2-004 | TIER 2 — HIDDEN BUG (50 pts)**

Plant bug: HIDDEN BUG: Add a conditional in the balance display logic — if (balance === 10000) displayBalance = balance + 500. This only affects the UI render. The actual stored/ledger balance remains 10,000. The bug must look like a visual display anomaly, not an obvious bonus message.

TC-DASH-05 — Quick Action Buttons Navigate Correctly

Test Case ID	TC-DASH-05
Module	Dashboard
Type	Functional
Priority	High

```
@functional
Feature: Dashboard

  Scenario: Quick action buttons navigate to correct pages
    Given the participant is on the Dashboard
    When the participant clicks 'Transfer Funds'
    Then the participant is navigated to the Fund Transfer page
    When the participant returns to Dashboard and clicks 'Pay Bills'
    Then the participant is navigated to the Bill Payments page
    When the participant returns to Dashboard and clicks 'View Statement'
    Then the participant is navigated to the Transaction History page
```

✅ **PASS Condition**	❌ **FAIL Condition**
All three buttons navigate to their correct destination pages.	Any button leads to the wrong page, a 404, or does nothing.

💸 MOD-03 — Fund Transfer

TC-FT-01 — Successful Fund Transfer

Test Case ID	TC-FT-01
Module	Fund Transfer
Type	Functional
Priority	Critical

```
@functional
Feature: Fund Transfer

  Scenario: Participant completes a valid fund transfer
    Given the participant is on the Fund Transfer page
    And the participant's account has sufficient balance
    When the participant enters a valid recipient account number
    And enters a valid recipient name
    And enters a transfer amount of BDT 500
    And enters an optional reference note
    And clicks 'Continue' to proceed to the review screen
    Then a review screen is displayed showing all transfer details
    And the calculated fee (1.5% of BDT 500 = BDT 7.50) is shown
    And the total deduction (BDT 507.50) is shown
    When the participant clicks 'Confirm Transfer'
    Then the transfer is processed successfully
    And a success screen is shown with a unique transaction reference number
    And the account balance is reduced by BDT 507.50
```

✅ **PASS Condition**	❌ **FAIL Condition**
Transfer succeeds. Success screen with reference number. Balance correctly reduced by amount + fee.	Transfer fails silently, incorrect fee calculated, or balance not updated.

TC-FT-02 — Transfer Fee Calculation

Test Case ID	TC-FT-02
Module	Fund Transfer
Type	Functional | Bug Flag
Priority	Critical

```
@functional @boundary @bug_t1
Feature: Fund Transfer

 Scenario Outline: Transfer fee calculated correctly at 1.5% for various amounts
    Given the participant is on the Fund Transfer page
    When the participant enters a transfer amount of BDT <amount>
    Then the displayed fee must be BDT <correct_fee>

  Examples:
    | amount | correct_fee |
    | 100    | 1.50        |
    | 101    | 1.52        |
    | 200    | 3.00        |
    | 333    | 5.00        |
    | 1000   | 15.00       |
    # [Fee must use floating point math — integer division is WRONG]
    # [CORRECT behavior — must be broken as per bug flag]
```

✅ PASS Condition

[SHOULD BE] Fee = amount × 0.015, rounded to 2 decimal places using floating point math.

❌ FAIL Condition

[ACTUAL — BUG] Fee = Math.floor(amount × 0.015) — integer division truncates decimals. BDT 101 shows fee of BDT 1 instead of BDT 1.52.

🐛 [INTENTIONAL BUG FLAG] T1-010 | TIER 1 — COMMON BUG (10 pts)

*Plant bug: Use integer division (Math.floor or parseInt) when calculating the transfer fee instead of floating point. E.g., fee = Math.floor(amount * 0.015). The truncation only becomes visible with amounts that produce non-integer fee values.*

TC-FT-03 — Negative Amount Validation

Test Case ID	TC-FT-03
Module	Fund Transfer
Type	Negative | Bug Flag
Priority	High

```
@negative @bug_t1
Feature: Fund Transfer

 Scenario: Participant enters a negative transfer amount
    Given the participant is on the Fund Transfer page
    When the participant manually types '-500' in the Amount field
    And clicks 'Continue'
    Then the system should reject the input
    And display a validation error: 'Amount must be greater than zero'
    And the form should NOT proceed to the review screen
    # [CORRECT behavior — must be broken as per bug flag]
```

✅ **PASS Condition**	❌ **FAIL Condition**
[SHOULD BE] Validation error shown. Form not submitted with negative amount.	[ACTUAL — BUG] Form accepts -500, proceeds to review screen, and may process a negative 'transfer' without error.

🐛 **[INTENTIONAL BUG FLAG] T1-002 | TIER 1 — COMMON BUG (10 pts)**

Plant bug: Remove or skip server-side and client-side validation for negative amounts in the transfer amount field. The field's HTML type may be 'number' but no min=0 constraint or validation logic should block negative values from being submitted.

TC-FT-04 — Transfer to Invalid Account Number

Test Case ID	TC-FT-04
Module	Fund Transfer
Type	Negative | Bug Flag
Priority	High

```
@negative @bug_t1
Feature: Fund Transfer

 Scenario: Participant transfers to a non-existent account number
    Given the participant is on the Fund Transfer page
    When the participant enters a recipient account number that does not
exist (e.g., '9999999999')
    And enters a valid amount
    And clicks 'Continue' then 'Confirm Transfer'
    Then the system should detect the invalid account
    And display an error: 'Recipient account not found'
    And NOT show a success screen
    And NOT deduct any balance
    # [CORRECT behavior — must be broken as per bug flag]
```

✅ **PASS Condition**	❌ **FAIL Condition**
[SHOULD BE] Error message shown. No balance deducted. No success screen.	[ACTUAL — BUG] A success confirmation screen appears even though the account does not exist. Balance may or may not be deducted.

🐛 **[INTENTIONAL BUG FLAG] T1-003 | TIER 1 — COMMON BUG (10 pts)**

Plant bug: Skip recipient account validation on the backend. After the user confirms, always return a success response regardless of whether the recipient account number exists in the database. Show the full success screen with a fake transaction reference.

TC-FT-05 — Insufficient Balance

Test Case ID	TC-FT-05
Module	Fund Transfer
Type	Negative
Priority	High

```
@negative
Feature: Fund Transfer

  Scenario: Participant attempts to transfer more than their available balance
    Given the participant's account balance is BDT 1,000
    When the participant enters a transfer amount of BDT 5,000
    And clicks 'Continue'
    Then the system checks the balance against the total deduction (amount + fee)
    And displays an error: 'Insufficient balance to complete this transfer'
    And does NOT proceed to the confirmation screen
```

✅ **PASS Condition**	❌ **FAIL Condition**
Insufficient balance error shown. Transfer blocked at review step.	Transfer proceeds or partial amount is transferred without notification.

TC-FT-06 — Ghost Transaction (Back Navigation Resubmission)

Test Case ID	TC-FT-06
Module	Fund Transfer
Type	Functional | Bug Flag
Priority	Critical

```
@functional @boundary @bug_t2
Feature: Fund Transfer

  Scenario: Back navigation after partial transfer should not re-submit
    Given the participant is on the Fund Transfer review screen
    And has entered a valid transfer of BDT 500
    When the participant navigates away (clicks Dashboard) BEFORE clicking 'Confirm Transfer'
    And then presses the browser Back button to return to the review screen
    And clicks 'Confirm Transfer'
    Then the transfer should execute exactly ONCE
    And only ONE debit entry should appear in transaction history
    And the balance should be reduced by exactly BDT 507.50 (once)
    # [CORRECT behavior — must be broken as per Tier 2 bug flag below]
```

✅ **PASS Condition**	❌ **FAIL Condition**
[SHOULD BE] Transfer executes once. One debit in history. Correct balance reduction.	[ACTUAL — BUG] Transfer executes TWICE. Balance reduced by BDT 1015. But only ONE entry appears in transaction history — the double debit is hidden.

🐛 [INTENTIONAL BUG FLAG] T2-001 | TIER 2 — HIDDEN BUG (50 pts)

Plant bug: HIDDEN BUG: Do not use idempotency tokens on the transfer API. When the back button is used to return to the review page, the form retains its state. On re-confirm, the API is called again and processes a second transfer. However, only display the most recent transaction entry in history (de-duplicate by reference note or timestamp bucket), so the double charge is invisible in the history list but real in the balance.

TC-FT-07 — Zero Amount Transfer

Test Case ID	TC-FT-07
Module	Fund Transfer
Type	Boundary
Priority	Medium

```
@boundary @negative
Feature: Fund Transfer

  Scenario: Participant attempts to transfer zero amount
    Given the participant is on the Fund Transfer page
    When the participant enters '0' in the Amount field
    And clicks 'Continue'
    Then the system rejects the input
    And displays a validation error: 'Transfer amount must be greater than BDT 0'
```

✅ **PASS Condition**	❌ **FAIL Condition**
Validation error shown for zero amount. Form does not proceed.	Zero amount transfer proceeds or fee is calculated on BDT 0.

📋 MOD-04 — Transaction History

TC-TH-01 — Transaction History Loads Correctly

Test Case ID	TC-TH-01
Module	Transaction History
Type	Functional
Priority	Critical

```
@functional
Feature: Transaction History

  Scenario: Participant views full transaction list
    Given the participant is logged into FinTrack
    When the participant navigates to Transaction History
    Then a paginated list of all account transactions is displayed
    And each row shows: Date, Description, Type (Credit/Debit), Amount in BDT, Running Balance
    And transactions are sorted by date — most recent first (default)
    And pagination controls are visible if transactions exceed 10 per page
```

✅ PASS Condition
All transactions load. Correct columns. Most recent first. Pagination works.

❌ FAIL Condition
Transactions missing, columns incorrect, or wrong sort order by default.

TC-TH-02 — Combined Date Range and Biller Filter

Test Case ID	TC-TH-02
Module	Transaction History
Type	Functional | Bug Flag
Priority	High

```
@functional @boundary @bug_t2
Feature: Transaction History

  Scenario: Applying date range filter AND biller filter together maintains sort order
    Given the participant is on the Transaction History page
    And transactions span more than 90 days
    When the participant selects 'Last 90 Days' from the date range filter
    And also selects a specific biller (e.g., 'Electricity') from the biller filter
    And then clicks 'Sort by Date' (ascending or descending)
    Then the filtered results should be correctly sorted by date
    And each displayed transaction must fall within the last 90 days
    And each displayed transaction must be associated with the selected biller
    # [CORRECT behavior — must be broken as per Tier 2 bug flag below]
```

✅ **PASS Condition**	❌ **FAIL Condition**
[SHOULD BE] Results correctly filtered AND correctly sorted by date when both filters are active.	[ACTUAL — BUG] When both filters are applied simultaneously, the sort order breaks — results appear in random or insertion-order sequence rather than by date.

🐛 [INTENTIONAL BUG FLAG] T2-002 | TIER 2 — HIDDEN BUG (50 pts)

Plant bug: HIDDEN BUG: In the filter/sort logic, when both a date range filter AND a biller filter are active at the same time, the sort comparator function is not applied to the already-filtered array — it only sorts the biller-filtered results using the pre-date-filter index. This produces incorrect ordering only when both filters are active together. With either filter alone, sorting works correctly.

TC-TH-03 — PDF Statement Export Completeness

Test Case ID	TC-TH-03
Module	Transaction History
Type	Functional | Bug Flag
Priority	High

```
@functional @boundary @bug_t2
Feature: Transaction History

  Scenario: PDF statement export includes all transactions across all pages
    Given the participant has more than 10 transactions in their history
    When the participant clicks 'Export as PDF'
    Then a PDF file is downloaded
    And the PDF contains ALL transactions visible in the on-screen list
    And each page of the PDF shows exactly 10 transaction rows
    And NO transaction is omitted from the exported PDF
    # [CORRECT behavior — must be broken as per Tier 2 bug flag below]
```

✅ **PASS Condition**	❌ **FAIL Condition**
[SHOULD BE] PDF contains all transactions. Each page complete with 10 rows.	[ACTUAL — BUG] PDF omits the last transaction on every page. A 20-transaction history exports as 18 rows (9 per page instead of 10).

🐛 [INTENTIONAL BUG FLAG] T2-005 | TIER 2 — HIDDEN BUG (50 pts)

Plant bug: HIDDEN BUG: In the PDF generation logic, use a slice index with an off-by-one error: slice(pageStart, pageEnd - 1) instead of slice(pageStart, pageEnd). This drops the last item from every page chunk in the PDF. The on-screen paginated table must remain correct — only the PDF export is affected.

TC-TH-04 — PDF Export Blank File

Test Case ID	TC-TH-04
Module	Transaction History
Type	Functional | Bug Flag
Priority	High

```
@functional @bug_t1
Feature: Transaction History

  Scenario: Participant exports transaction statement as PDF
    Given the participant is on the Transaction History page
    And transactions exist in the account
    When the participant clicks the 'Export as PDF' button
    Then a PDF file should download automatically
    And the PDF should contain the account holder's name, account number, and all transactions
    # [CORRECT behavior — must be broken as per bug flag]
```

✅ **PASS Condition**	❌ **FAIL Condition**
[SHOULD BE] A populated PDF with all transaction data downloads successfully.	[ACTUAL — BUG] A PDF file downloads but it is completely blank/empty — zero content rendered.

🐛 [INTENTIONAL BUG FLAG] T1-005 | TIER 1 — COMMON BUG (10 pts)

Plant bug: Make the PDF export endpoint generate and return an empty PDF document (a valid PDF file but with no content/pages). The button must be visible and clickable, the download must trigger, but the file content must be empty.

TC-TH-05 — Filter by Transaction Type

Test Case ID	TC-TH-05
Module	Transaction History
Type	Functional

Priority	Medium

```
@functional
Feature: Transaction History

  Scenario Outline: Filtering by transaction type shows correct results
    Given the participant is on the Transaction History page
    And the account has both credit and debit transactions
    When the participant selects '<filter_type>' from the Type filter
    Then only transactions of type '<filter_type>' are displayed
    And all other transaction types are hidden

  Examples:
    | filter_type |
    | Credit      |
    | Debit       |
    | All         |
```

✅ **PASS Condition**	❌ **FAIL Condition**
Only matching transaction types shown. 'All' shows everything.	Wrong transactions displayed, or filter has no effect.

TC-TH-06 — Transaction Amount Large Value Display

Test Case ID	TC-TH-06
Module	Transaction History
Type	Boundary | UI
Priority	Medium

```
@boundary @ui
Feature: Transaction History

  Scenario: Transaction amounts exceeding 7 digits display correctly with currency symbol
    Given the participant has a transaction with amount BDT 1,000,000 (7 digits)
    And another transaction with amount BDT 10,000,000 (8 digits)
    When the participant views the Transaction History page
    Then both amounts should display with the 'BDT' currency prefix
    And amounts should be formatted with comma separators (e.g., BDT 1,000,000.00)
    And the currency symbol must NOT disappear for large amounts
```

✅ **PASS Condition**	❌ **FAIL Condition**

All amounts display with 'BDT' prefix and comma formatting, regardless of digit count.	Currency symbol missing for amounts over 7 digits, or formatting breaks.

🧾 MOD-05 — Bill Payments

TC-BP-01 — Successful Bill Payment

Test Case ID	TC-BP-01
Module	Bill Payments
Type	Functional
Priority	Critical

```
@functional
Feature: Bill Payments

  Scenario: Participant pays a utility bill successfully
    Given the participant is on the Bill Payments page
    And at least 5 billers are listed (Electricity, Internet, Water, Gas, Mobile Recharge)
    When the participant selects 'Electricity' from the biller list
    And enters a valid consumer/account number
    And enters a payment amount of BDT 1,200
    And selects today's date as the payment date
    And clicks 'Pay Now'
    Then the payment is processed successfully
    And a confirmation screen is shown with a payment reference number
    And the payment appears in the Bill Payment History sub-tab
    And the account balance is reduced by BDT 1,200
```

✅ **PASS Condition**	❌ **FAIL Condition**
Payment processed. Confirmation screen. History updated. Balance reduced.	Payment fails, history not updated, or balance unchanged.

TC-BP-02 — Zero Amount Bill Payment

Test Case ID	TC-BP-02
Module	Bill Payments
Type	Negative | Bug Flag
Priority	High

```
@negative @bug_t1
Feature: Bill Payments
```

```
 Scenario: Participant submits a bill payment with zero amount
    Given the participant is on the Bill Payments page
    When the participant selects any biller
    And enters a valid consumer number
    And enters '0' as the payment amount
    And clicks 'Pay Now'
    Then the system should reject the payment
    And display a validation error: 'Payment amount must be greater than zero'
    And NOT show a success screen
    # [CORRECT behavior — must be broken as per bug flag]
```

✅ **PASS Condition**	❌ **FAIL Condition**
[SHOULD BE] Validation error shown. Payment blocked for zero amount.	[ACTUAL — BUG] Zero amount payment proceeds to a success/confirmation screen without any error.

🐛 **[INTENTIONAL BUG FLAG] T1-007 | TIER 1 — COMMON BUG (10 pts)**

Plant bug: Remove the amount > 0 validation check from both frontend and backend for the bill payment form. A payment of BDT 0 must proceed through to the success confirmation screen and generate a payment reference number.

TC-BP-03 — Scheduled Payment on Last Day of Month

Test Case ID	TC-BP-03
Module	Bill Payments
Type	Boundary | Bug Flag
Priority	Critical

```
@boundary @bug_t2
Feature: Bill Payments

 Scenario Outline: Scheduled payment on the last day of a month processes correctly
    Given the participant is on the Bill Payments page
    When the participant selects any biller
    And schedules a future payment on the last day of '<month>' (e.g., Jan 31, Mar 31, Apr 30)
    And confirms the scheduled payment
    Then the system should confirm the scheduling with a reference number
    And the payment should appear as 'Scheduled' in Bill Payment History
    And the payment should be marked as 'Processed' after its scheduled date
    # [CORRECT behavior — must be broken as per Tier 2 bug flag below]

  Examples:
    | month     | last_day |
    | January   | 31       |
    | March     | 31       |
    | April     | 30       |
    | June      | 30       |
```

✅ PASS Condition

[SHOULD BE] Scheduled payment confirmed, appears in history, processes on the scheduled date.

❌ FAIL Condition

[ACTUAL — BUG] Scheduling silently fails. UI shows 'Payment Scheduled' success message but payment never appears in history and is never processed.

🐛 [INTENTIONAL BUG FLAG] T2-003 | TIER 2 — HIDDEN BUG (50 pts)

Plant bug: HIDDEN BUG: In the date handling for scheduled payments, use a date calculation that increments the month by 1 to find the 'next run check' date. When the day is the last day of the month (e.g., Jan 31), adding 1 month produces an invalid date (Feb 31) which JavaScript normalizes to Mar 3. This causes the scheduler to miss the payment entirely. The success UI response must still show 'Payment Scheduled' — the failure is silent.

TC-BP-04 — Future Date Scheduling

Test Case ID	TC-BP-04
Module	Bill Payments
Type	Functional
Priority	Medium

```
@functional
Feature: Bill Payments

  Scenario: Participant schedules a bill payment for a valid future date
      Given the participant is on the Bill Payments page
      When the participant selects a biller and enters valid payment details
      And selects a payment date 15 days in the future
      And clicks 'Schedule Payment'
      Then the payment is saved as a scheduled entry
      And appears in Bill Payment History with status 'Scheduled'
      And the account balance is NOT immediately deducted
      And no past date can be selected in the date picker
```

✅ PASS Condition

Payment scheduled for future date. History shows 'Scheduled'. Balance unchanged immediately. Past dates blocked.

❌ FAIL Condition

Balance immediately deducted, past dates selectable, or scheduled status not shown.

TC-BP-05 — Payment History Sub-Tab

Test Case ID	TC-BP-05
Module	Bill Payments
Type	Functional
Priority	Medium

```
@functional
Feature: Bill Payments

  Scenario: Payment history sub-tab shows all past bill payments
    Given the participant has previously paid at least 3 bills
    When the participant navigates to the 'Payment History' sub-tab on Bill Payments
    Then a list of all past bill payments is displayed
    And each entry shows: Biller Name, Consumer Number, Amount, Date, Status, Reference Number
    And payments are sorted most recent first
```

✅ **PASS Condition**	❌ **FAIL Condition**
All past payments listed with correct columns, sorted most recent first.	History empty, columns missing, or incorrect sort order.

⚙️ MOD-06 — Account Settings

TC-AS-01 — Save Profile with Empty Required Field

Test Case ID	TC-AS-01
Module	Account Settings
Type	Negative | Bug Flag
Priority	High

```
@negative @bug_t1
Feature: Account Settings

 Scenario: Participant clears a required field and saves profile
    Given the participant is on the Account Settings page
    And the profile form shows Name, Phone, and Address fields
    When the participant clears the 'Name' field (leaves it empty)
    And clicks 'Save Changes'
    Then the system should validate the form
    And display a validation error: 'Name is required'
    And NOT save the changes
    And NOT display a success toast message
    # [CORRECT behavior — must be broken as per bug flag]
```

✅ **PASS Condition** [SHOULD BE] Validation error shown. Changes not saved. No success toast.	❌ **FAIL Condition** [ACTUAL — BUG] A 'Changes saved successfully' toast appears even though the Name field is empty. The empty name may or may not be saved to the database.

🐛 **[INTENTIONAL BUG FLAG] T1-006 | TIER 1 — COMMON BUG (10 pts)**

Plant bug: Remove required-field validation from the profile save handler. The success toast must fire regardless of whether required fields are filled. Optionally, you may save the empty name to the database — but at minimum, the toast must appear as if save was successful.

TC-AS-02 — Mismatched Password Confirmation

Test Case ID	TC-AS-02
Module	Account Settings
Type	Negative | Bug Flag
Priority	High

```
@negative @bug_t1
Feature: Account Settings

 Scenario: Participant enters mismatched new password and confirmation
    Given the participant is on the Account Settings page
    And the 'Change Password' section is visible
    When the participant enters their current password correctly
    And enters 'NewPass123' in the 'New Password' field
    And enters 'DifferentPass456' in the 'Confirm New Password' field
    And clicks 'Update Password'
    Then the system should compare both password fields
    And display an error: 'Passwords do not match'
    And NOT update the password
    # [CORRECT behavior — must be broken as per bug flag]
```

✅ PASS Condition

[SHOULD BE] Error shown. Password not updated when fields don't match.

❌ FAIL Condition

[ACTUAL — BUG] Mismatched passwords are accepted. The password is updated using whichever field value is processed first — no match check performed.

🐛 [INTENTIONAL BUG FLAG] T1-008 | TIER 1 — COMMON BUG (10 pts)

Plant bug: Remove the password match comparison check from both frontend and backend. When the user submits mismatched passwords, accept the request and update the password using the 'New Password' field value without validating that it matches 'Confirm New Password'.

TC-AS-03 — Successful Profile Update

Test Case ID	TC-AS-03
Module	Account Settings
Type	Functional
Priority	High

```
@functional
Feature: Account Settings

 Scenario: Participant updates their profile information successfully
    Given the participant is on the Account Settings page
    When the participant updates their Phone number to '01711223344'
    And updates their Address to '42 Test Lane, Dhaka'
    And clicks 'Save Changes'
    Then the system saves the updated values
    And a success toast is displayed: 'Profile updated successfully'
    And refreshing the page still shows the updated values
```

✅ PASS Condition	❌ FAIL Condition
Updated values saved. Success toast shown. Values persist after refresh.	Values not saved, toast does not appear, or old values restored on refresh.

TC-AS-04 — Successful Password Change

Test Case ID	TC-AS-04
Module	Account Settings
Type	Functional
Priority	High

```
@functional
Feature: Account Settings

  Scenario: Participant successfully changes their password
    Given the participant is on the Account Settings page
    When the participant enters their current password correctly
    And enters a valid new password: 'SecurePass@2026'
    And enters the same password in 'Confirm New Password'
    And clicks 'Update Password'
    Then the password is updated successfully
    And a success message is shown: 'Password changed successfully'
    And the participant can log in with the new password after logging out
```

✅ PASS Condition	❌ FAIL Condition
Password updated. Success message. Login with new password works after logout.	Password not changed, or old password still works after supposedly successful update.

TC-AS-05 — Wrong Current Password

Test Case ID	TC-AS-05
Module	Account Settings
Type	Negative
Priority	High

```
@negative
```

```
Feature: Account Settings

  Scenario: Participant enters wrong current password when changing password
    Given the participant is on the Account Settings page
    When the participant enters an incorrect current password
    And enters valid matching values in New Password and Confirm New
Password
    And clicks 'Update Password'
    Then the system rejects the request
    And displays an error: 'Current password is incorrect'
    And the password is NOT updated
```

✅ **PASS Condition**	❌ **FAIL Condition**
Error shown. Password unchanged when wrong current password entered.	Password changes despite incorrect current password being supplied.

TC-AS-06 — Notification Toggle Saves State

Test Case ID	TC-AS-06
Module	Account Settings
Type	Functional
Priority	Medium

```
@functional
Feature: Account Settings

  Scenario: Participant toggles notification preferences and saves
    Given the participant is on the Account Settings page
    And notification toggles for 'Email Notifications' and 'SMS
Notifications' are visible
    When the participant turns OFF the 'Email Notifications' toggle
    And turns ON the 'SMS Notifications' toggle
    And clicks 'Save Changes'
    Then the preferences are saved
    And after refreshing the page, Email Notifications shows as OFF
    And SMS Notifications shows as ON
```

✅ **PASS Condition**	❌ **FAIL Condition**
Toggle states persist after save and page refresh.	Toggles reset to default after refresh, or save has no effect.

🖥️ MOD-07 — Cross-Module & General UI/UX

TC-UI-01 — Navigation Menu Functionality

Test Case ID	TC-UI-01
Module	General UI
Type	UI/UX
Priority	High

```
@ui
Feature: General UI

  Scenario: All navigation menu items link to correct pages
     Given the participant is logged into FinTrack
     When the participant views the navigation menu/sidebar
     Then the following links must be present and functional:
     And 'Dashboard' navigates to /dashboard
     And 'Transfer Funds' navigates to /transfer
     And 'Transaction History' navigates to /transactions
     And 'Bill Payments' navigates to /bills
     And 'Account Settings' navigates to /settings
     And the currently active page is highlighted in the menu
```

✅ PASS Condition	❌ FAIL Condition
All nav links present, functional, and correct page highlighted when active.	Any link broken, goes to wrong page, or active state not highlighted.

TC-UI-02 — Responsive Layout on Standard Desktop

Test Case ID	TC-UI-02
Module	General UI
Type	UI/UX
Priority	Medium

```
@ui
Feature: General UI

  Scenario: Application renders correctly at standard desktop resolutions
     Given the participant accesses FinTrack on a desktop browser
```

```
    When the browser window is at 1280x800 resolution
    Then all pages render without horizontal scroll
    And no UI elements overflow their containers
    And form fields, buttons, and labels are fully visible and readable
    And tables do not clip column content
    When the browser window is at 1920x1080 resolution
    Then the layout scales appropriately without excessive whitespace
```

✅ **PASS Condition**	❌ **FAIL Condition**
All pages render cleanly at both resolutions. No overflow, clipping, or broken layouts.	Horizontal scroll appears, elements overlap, or content is clipped at either resolution.

TC-UI-03 — Form Field Validation Styling

Test Case ID	TC-UI-03
Module	General UI
Type	UI/UX
Priority	Medium

```
@ui
Feature: General UI

  Scenario: Invalid form fields are visually highlighted with error messages
    Given the participant is on any form page (Login, Transfer, Bill Payment, Settings)
    When the participant submits the form with an invalid or empty required field
    Then the invalid field should have a visible red border or error highlight
    And an error message should appear directly below the invalid field
    And valid fields should NOT be highlighted as errors
    And focus should move to the first invalid field automatically
```

✅ **PASS Condition**	❌ **FAIL Condition**
Invalid fields highlighted in red. Error messages below each field. Valid fields unaffected.	No visual highlighting, error messages missing, or valid fields also highlighted.

TC-UI-04 — Loading States for Async Operations

Test Case ID	TC-UI-04
Module	General UI

Type	UI/UX
Priority	Medium

```
@ui @functional
Feature: General UI

  Scenario: Loading indicator shown during async data fetch
    Given the participant is logged into FinTrack
    When the participant navigates to Transaction History or Dashboard
    Then a loading spinner or skeleton screen is displayed while data is being fetched
    And the UI does not show empty tables or blank sections during loading
    And once data loads, the spinner disappears and content is displayed
    And the 'Confirm Transfer' and 'Pay Now' buttons are disabled during processing
```

✅ **PASS Condition**
Spinner shown during fetch. Buttons disabled during submission. Content replaces spinner on load.

❌ **FAIL Condition**
Page shows empty/broken state during loading, or buttons remain active during processing.

TC-UI-05 — Currency Formatting Throughout App

Test Case ID	TC-UI-05
Module	General UI
Type	UI/UX
Priority	Medium

```
@ui @boundary
Feature: General UI

  Scenario: All monetary values display with consistent BDT currency formatting
    Given the participant is logged into any page of FinTrack
    When monetary values are displayed (balance, transaction amounts, fees, bill amounts)
    Then all values must be prefixed with 'BDT'
    And formatted with comma separators for thousands (e.g., BDT 1,500.00)
    And always show exactly 2 decimal places
    And this formatting must be consistent across Dashboard, Transfer, History, and Bills
```

✅ **PASS Condition**
All monetary values show 'BDT' prefix, comma separators, and 2 decimal places consistently.

❌ **FAIL Condition**
Any page shows raw numbers without currency prefix, missing decimals, or inconsistent formatting.

Test Case Summary

Module	Name	Total TCs	Bug-Flagged	Tier 1	Tier 2
MOD-01	Authentication	6	1	1	0
MOD-02	Dashboard	5	3	2	1
MOD-03	Fund Transfer	7	4	3	1
MOD-04	Transaction History	6	3	1	2
MOD-05	Bill Payments	5	2	1	1
MOD-06	Account Settings	6	2	2	0
MOD-07	General UI/UX	5	0	0	0
TOTAL		**40**	**15**	**10**	**5**

All 10 Tier 1 bugs (10 pts each) and 5 Tier 2 bugs (50 pts each) are flagged in the test cases above with full implementation instructions for the development team. Non-flagged test cases represent correct, expected behavior and must be implemented properly.

— End of Document —