



**Dhirubhai Ambani  
University  
Technology**

Formerly DA-IICT

## GROUP - 28

SYSTEM TESTING – ChatBot and Profile  
Testers : Kirtan Chauhan and Ruchir Joshi

## Introduction

This document presents the System Testing approach for the AI-Assistant (CHATBOT) and Profile modules of the Smart Expense application. The objective is to validate user-facing functionality using Black-Box Testing, focusing on input–output behaviour without inspecting the internal code.

Both Equivalence Class Partitioning (ECP) and Boundary Value Analysis (BVA) techniques are applied to systematically identify valid and invalid input categories.

Additionally, Acceptance Testing is conducted to verify that the system meets user requirements by executing real inputs on the interface and observing the actual application response.

# BlackBox Testing — AI-Assistant

## User-Visible Constraints

### 1. Message Input Field

- Cannot be empty
- Cannot contain only spaces
- Accepts any text characters

### 2. Send Button

- Disabled when input is empty
- Disabled when bot is typing
- Enabled only when input has valid non-whitespace text

### 3. Quick Actions

- Clicking sends predefined messages automatically
- Should work even when input box is empty
- Should be disabled when bot is typing

### 4. Suggestions

- Only appear after bot messages
- Are clickable
- Auto-send the suggestion text

### 5. Typing Indicator

- Appears when bot is "thinking"

- During this time:
  - Input box is disabled
  - Send button is disabled

#### 6. Chat Flow

- User messages appear on the right
- Bot messages appear on the left
- Chat auto-scrolls to the newest message

#### 7. Authentication

- If user is not logged in → they cannot send messages (UI prevents interaction)

#### 8. Markdown Rendering

- Bot messages may include bullet points or formatted text
- The UI must properly render markdown

## 2. Equivalence Class Partitioning (ECP)

| <b>EC</b> | <b>Input</b>       | <b>Chat State</b> | <b>Expected Outcome</b>      |
|-----------|--------------------|-------------------|------------------------------|
| E1        | Valid text         | bot not typing    | Message sent + bot response  |
| E2        | Empty input        | bot not typing    | Cannot send                  |
| E3        | Spaces only        | bot not typing    | Cannot send                  |
| E4        | Valid text         | bot typing        | Cannot send                  |
| E5        | Quick Action       | bot not typing    | Auto-send + bot response     |
| E6        | Suggestion click   | bot not typing    | Auto-send + bot response     |
| E7        | Quick Action       | bot typing        | Blocked (no send)            |
| E8        | User not logged in | any               | Cannot send message          |
| E9        | Valid text         | bot not typing    | Bot returns markdown content |

### 3. Boundary Value Analysis (BVA)

| Test | Input                          | Expected Output | Reason                  |
|------|--------------------------------|-----------------|-------------------------|
| BVA1 | ""                             | fail            | Input cannot be empty   |
| BVA2 | " "                            | fail            | Whitespace-only invalid |
| BVA3 | "a" (minimum valid input)      | success         | Minimum valid           |
| BVA4 | Very long message (300+ chars) | success         | Stress test             |
| BVA5 | Quick Action click             | success         | Should auto-send        |
| BVA6 | Suggestion click               | success         | Should auto-send        |
| BVA7 | Message typed while bot typing | fail            | Send blocked            |
| BVA8 | Bot returns markdown           | success         | UI must render markdown |
| BVA9 | User logged out                | fail            | System blocks sending   |

### Acceptance Testing — AI Assistant Chatbot

#### Test 1 — Sending a Valid Message

Input:

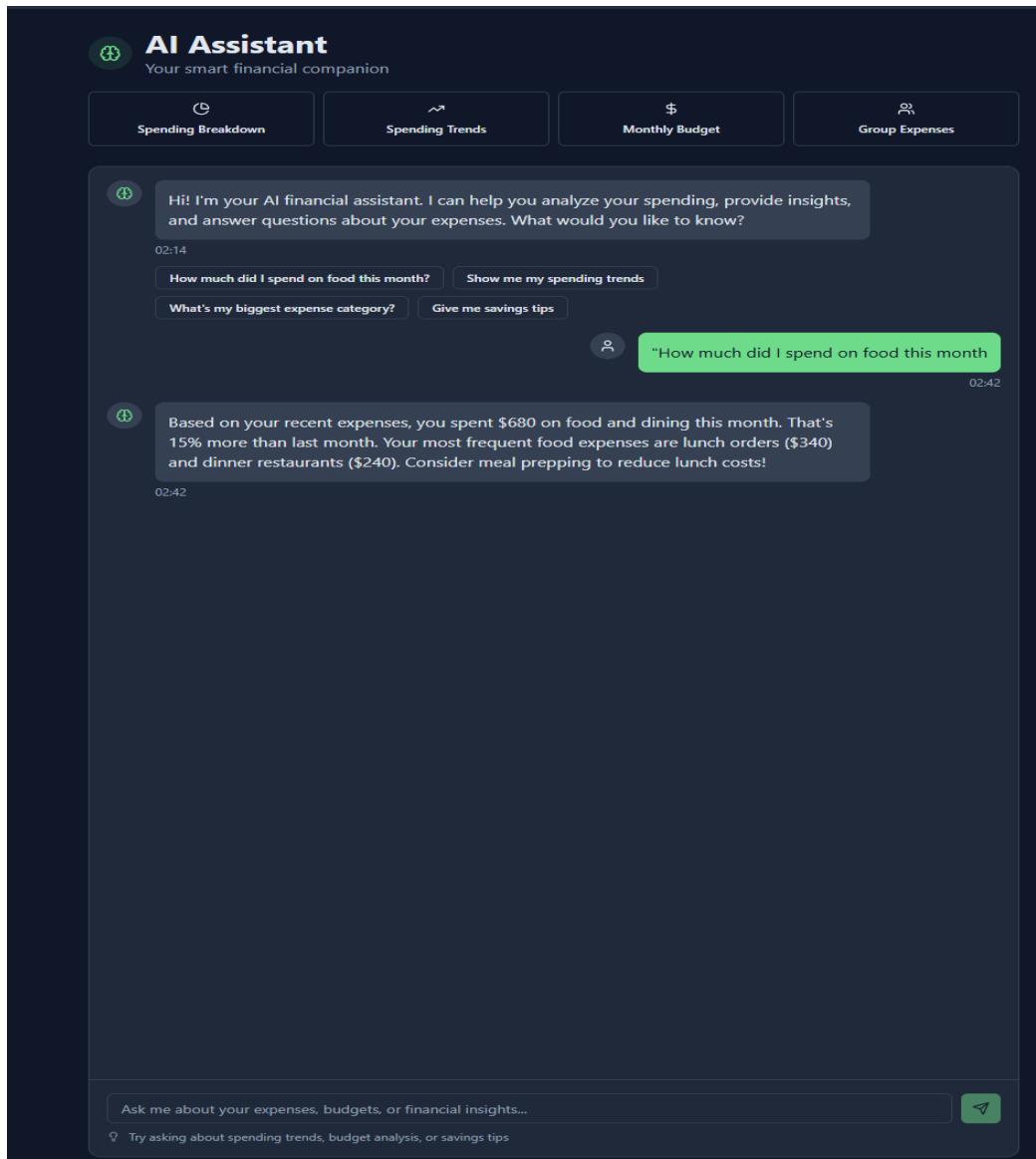
"How much did I spend on food this month?"

Action:

Type the message and press Send.

Expected Output:

Bot responds with a spending breakdown message related to food.



## Test 2 — Sending an Empty Message

Input:

"" (no text)

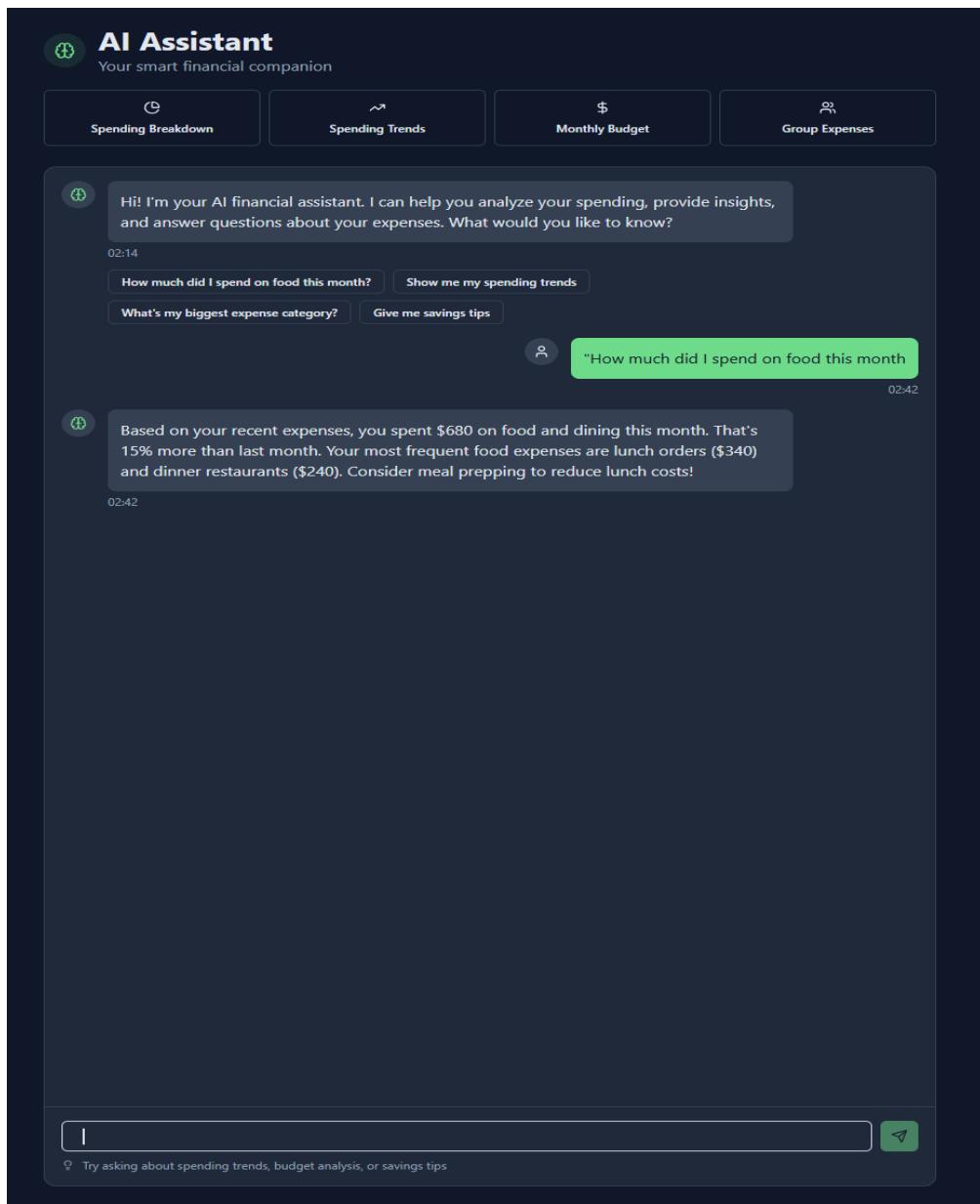
Action:

Attempt to press Send.

Expected Output:

Message is NOT sent.

Send button remains disabled.



### Test 3 — Quick Action Button

Input:

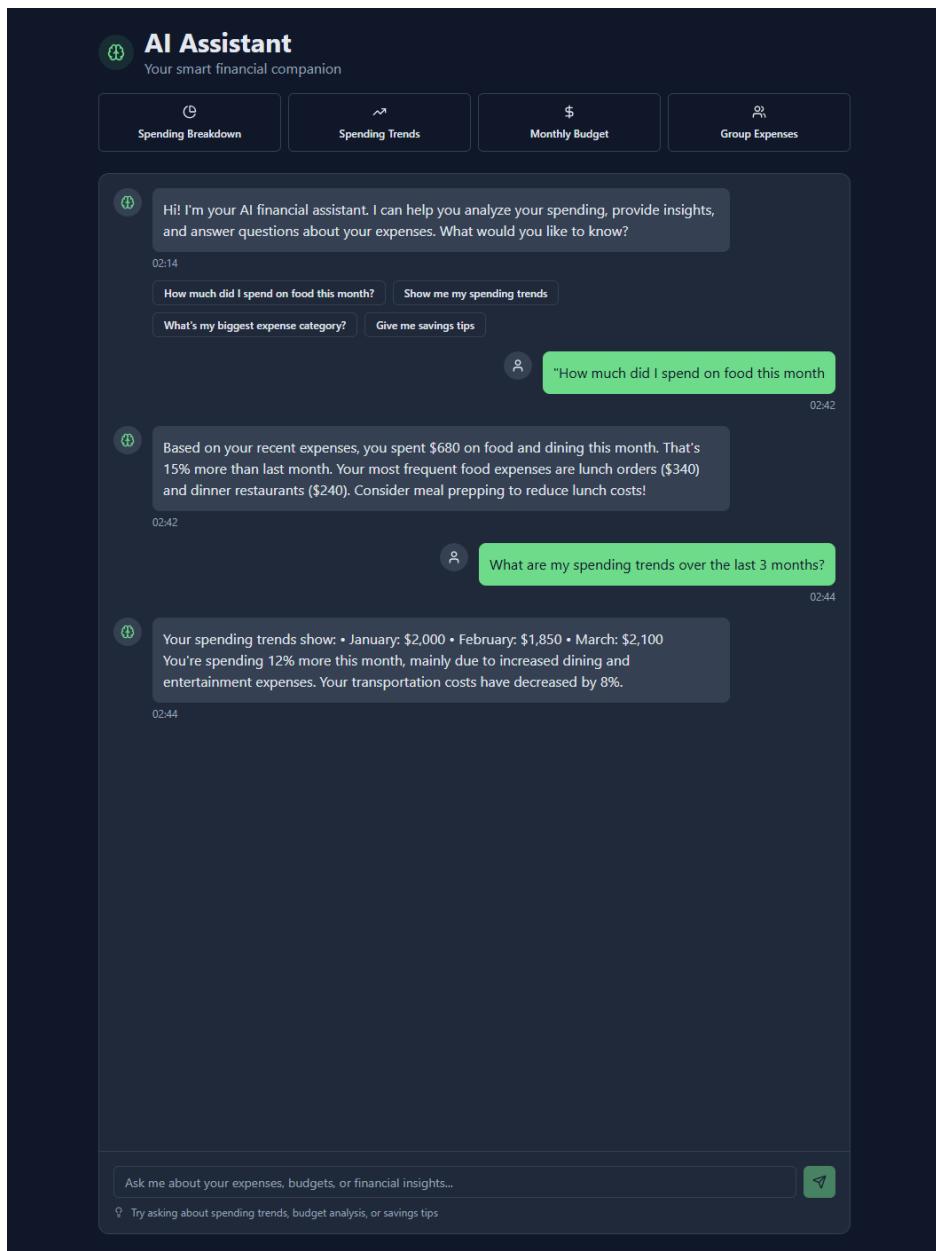
Click “Spending Trends” quick action card.

Action:

User does not type anything; clicks the Quick Action button.

Expected Output:

The Quick Action query auto-sends, and a bot response appears with trend analysis.



## Test 4 — Suggestion Chip Click

Input:

Click bot suggestion:

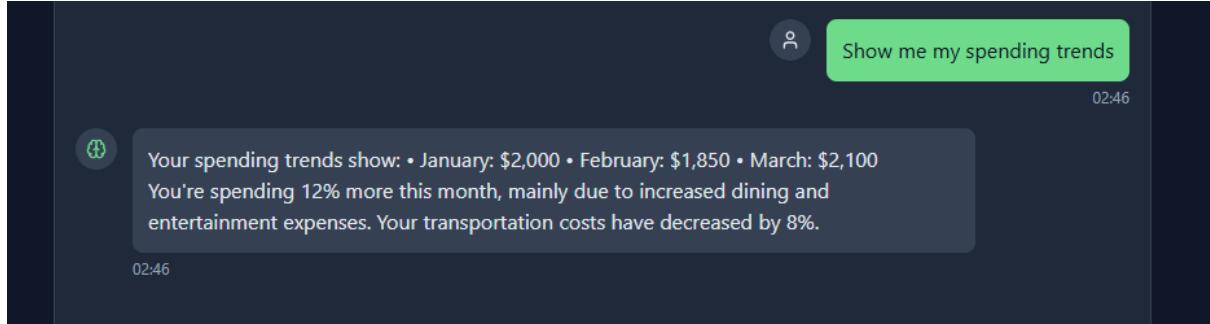
“Show me my spending trends”

Action:

Click the suggested chip under the bot message.

## Expected Output:

Message is auto-sent, bot responds with formatted markdown text.



## Test 5 — User Tries Sending While Bot is Typing

### Input:

Type “Hello” while bot is typing.

### Action:

Try clicking Send during the typing indicator.

### Expected Output:

Send button is disabled.

Message cannot be sent until bot reply finishes.



## Black-Box Testing — Profile Page

### 1. Field Constraints (User-Visible Only)

#### A. Name Field

- Cannot be empty
- Must contain alphabetic characters
- Should allow alphanumeric (kirtan123) — UI doesn't restrict

- No minimum or maximum shown, but assume:
  - Should not exceed 50 characters (standard usability assumption)

#### B. Phone Number Field

- Optional
- If entered:
  - Must be valid phone format (digits only)
  - Should typically be 10–15 digits
  - No alphabetic characters allowed

#### C. Profile Picture Upload

- Optional
- If uploaded:
  - Must be an image (JPG, PNG) — inferred from typical systems
  - Invalid files should be rejected

#### D. Save Changes Button

Allowed only when:

- At least one editable field changes
- Inputs meet constraints

#### E. Cancel Button

- Discards all changes
- Returns fields to previous values

## 2. Equivalence Class Partitioning (ECP)

| EC | Name  | Phone   | Profile Pic  | Expected Result |
|----|-------|---------|--------------|-----------------|
| E1 | valid | valid   | valid/none   | success         |
| E2 | valid | valid   | invalid file | error           |
| E3 | valid | invalid | valid/none   | error           |

| <b>EC</b>  | <b>Name</b>            | <b>Phone</b>               | <b>Profile Pic</b> | <b>Expected Result</b> |
|------------|------------------------|----------------------------|--------------------|------------------------|
| <b>E4</b>  | invalid (empty)        | valid                      | valid/none         | error                  |
| <b>E5</b>  | invalid (symbols only) | valid                      | valid/none         | error                  |
| <b>E6</b>  | valid                  | empty (allowed)            | valid/none         | success                |
| <b>E7</b>  | valid                  | invalid format (alphabets) | valid/none         | error                  |
| <b>E8</b>  | no changes             | —                          | —                  | Save disabled          |
| <b>E9</b>  | valid                  | excessive digits (>15)     | valid/none         | error                  |
| <b>E10</b> | valid                  | valid                      | none               | success                |

### **3. Boundary Value Analysis (BVA)**

| <b>Field</b> | <b>Test Input</b> | <b>Expected Output</b> | <b>Reason</b>        |
|--------------|-------------------|------------------------|----------------------|
| <b>Name</b>  | "" (empty)        | fail                   | Name cannot be empty |
| <b>Name</b>  | "a" (1 char)      | pass                   | Minimum valid        |

| Field        | Test Input                      | Expected Output        | Reason                         |
|--------------|---------------------------------|------------------------|--------------------------------|
| Name         | Very long (51+ chars)           | fail                   | Exceeds typical UI constraints |
| Name         | "!!!"                           | fail                   | No alphabetic characters       |
| Phone        | ""                              | pass                   | Optional field                 |
| Phone        | "1234567890" (10 digits)        | pass                   | Valid                          |
| Phone        | "12345" (too short)             | fail                   | Invalid phone format           |
| Phone        | "1234567890123456" (>15 digits) | fail                   | Exceeds phone length           |
| Phone        | "abcd1234"                      | fail                   | Alphabets not allowed          |
| Profile Pic  | No file                         | pass                   | Optional                       |
| Profile Pic  | JPG/PNG                         | pass                   | Valid format                   |
| Profile Pic  | PDF/EXE                         | fail                   | Invalid type                   |
| Save Changes | No edits made                   | fail (button disabled) | Must change something          |

## Acceptance Testing — Profile Page

### Test 1 — Update Name Successfully

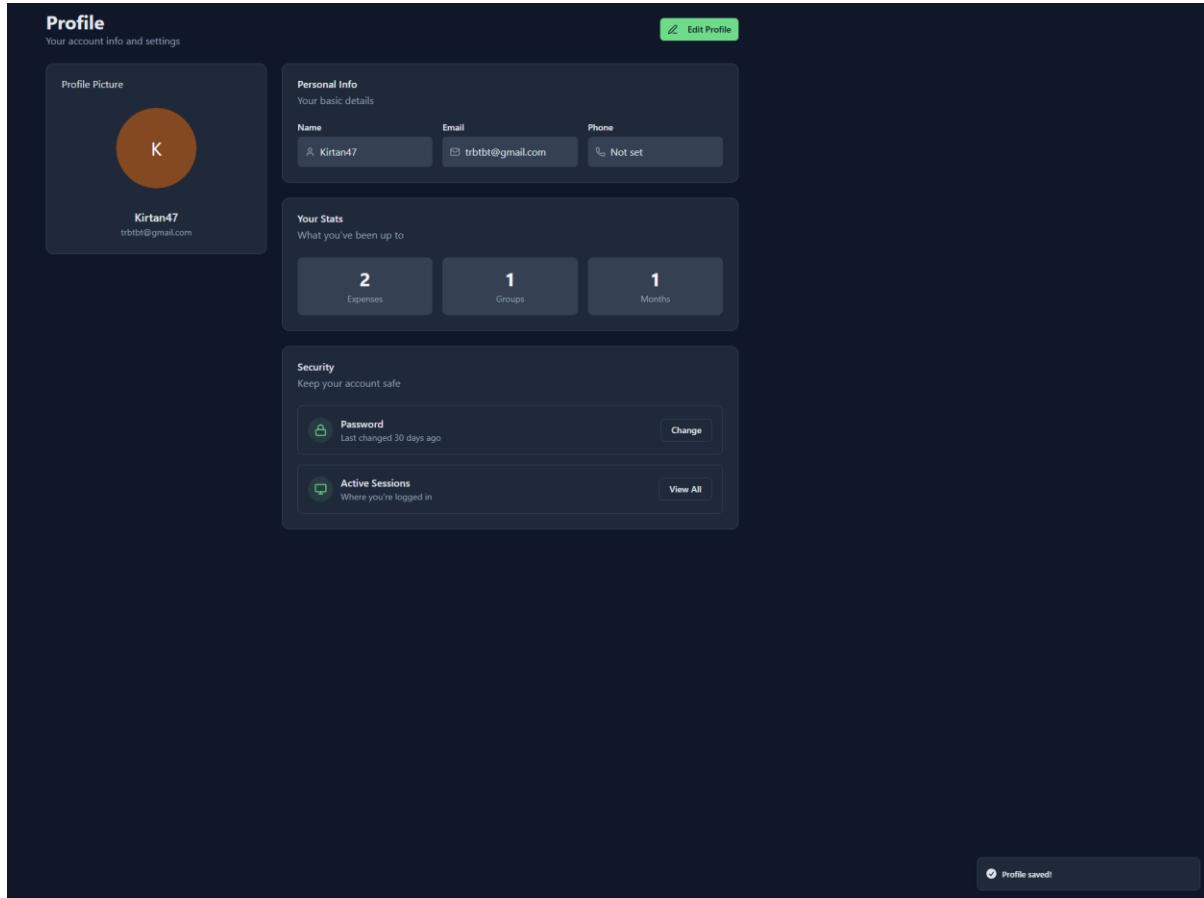
**Action:**

Click **Edit Profile**, change Name → “kirtan47”, then click **Save Changes**.

**Expected Output:**

Name is updated and reflected on the Profile page.

Success message



---

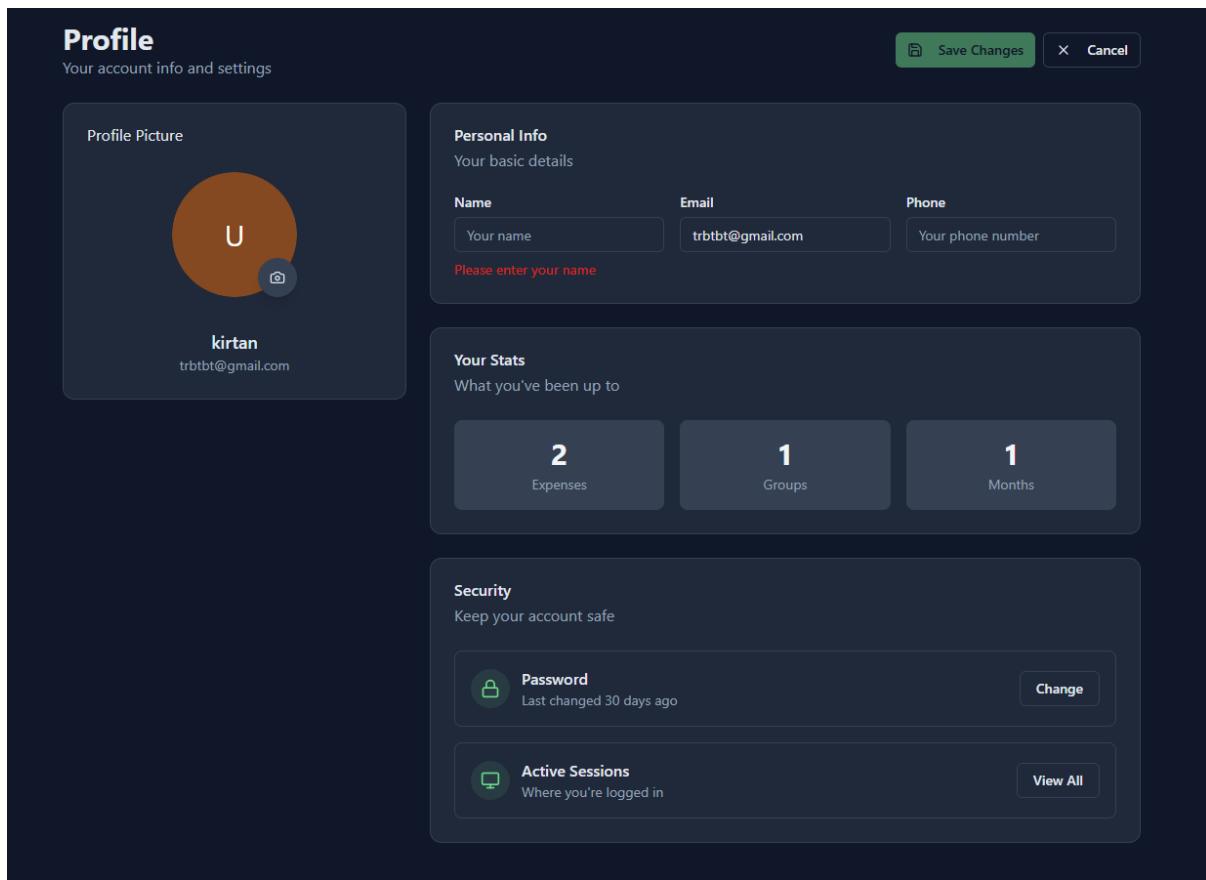
**Test 2 — Attempt to Save Empty Name****Action:**

Click **Edit Profile**, clear Name field completely, click **Save Changes**.

**Expected Output:**

Save should **fail**.

Error message or Save button disabled (depending on implementation).



### Test 3 — Enter Invalid Phone Number (Alphabetic Characters)

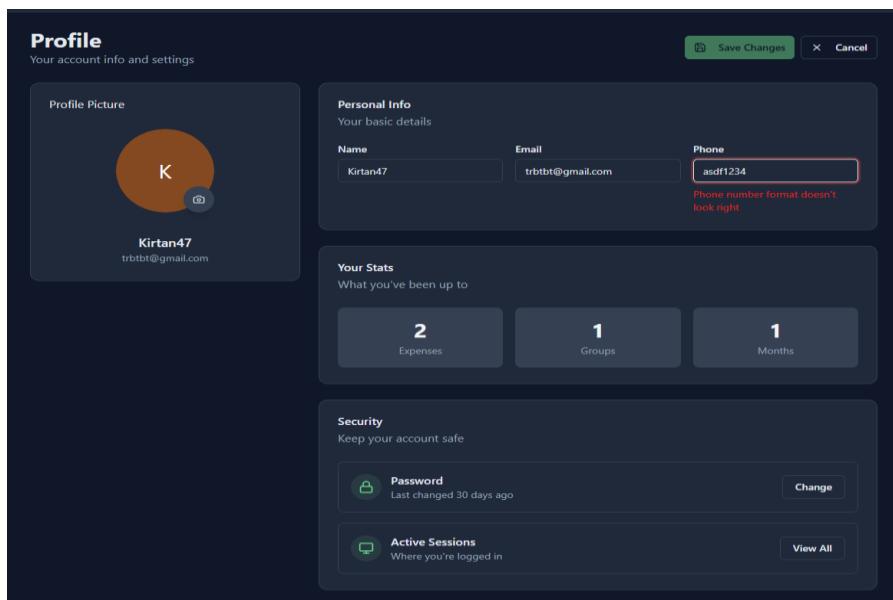
#### Action:

Click **Edit Profile**, enter phone: "abcd123", click **Save Changes**.

#### Expected Output:

Save should fail.

Phone field should show validation error or Save should be blocked.



#### **Test 4 — Upload Invalid Profile Picture Format**

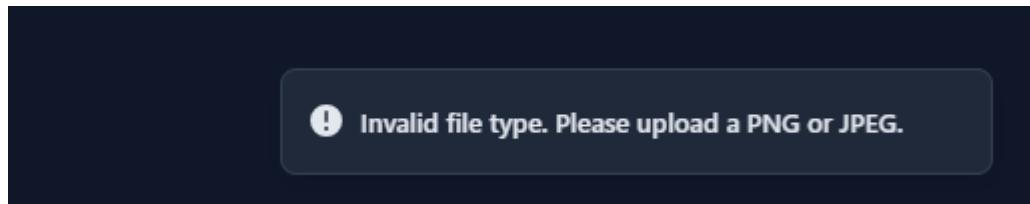
##### **Action:**

Click the camera icon on Profile Picture → upload file.pdf or file.exe.

##### **Expected Output:**

Image upload should **fail**.

Error shown OR file not accepted.



---

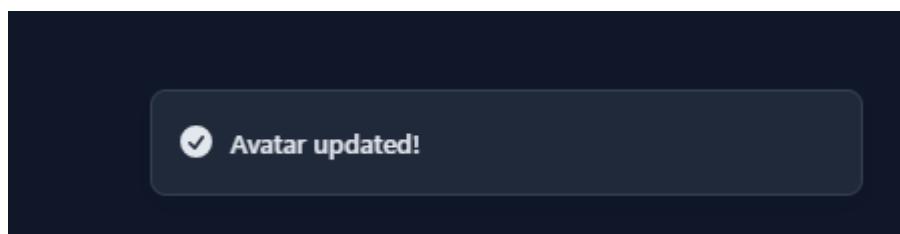
#### **Test 5 — Upload Valid Profile Picture**

##### **Action:**

Click profile pic → upload a valid JPG/PNG image.

##### **Expected Output:**

Profile picture updates instantly or after Save.



---

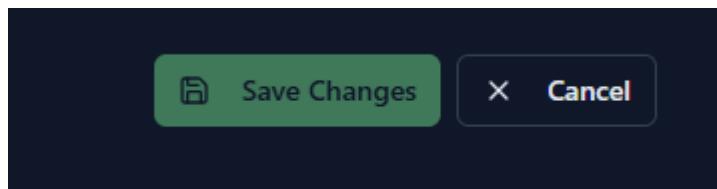
#### **Test 6 — Save Without Making Any Changes**

##### **Action:**

Open Profile → click **Edit Profile** → do not change anything → click **Save Changes**.

##### **Expected Output:**

Save should be **blocked** (button disabled) or no update made.



---

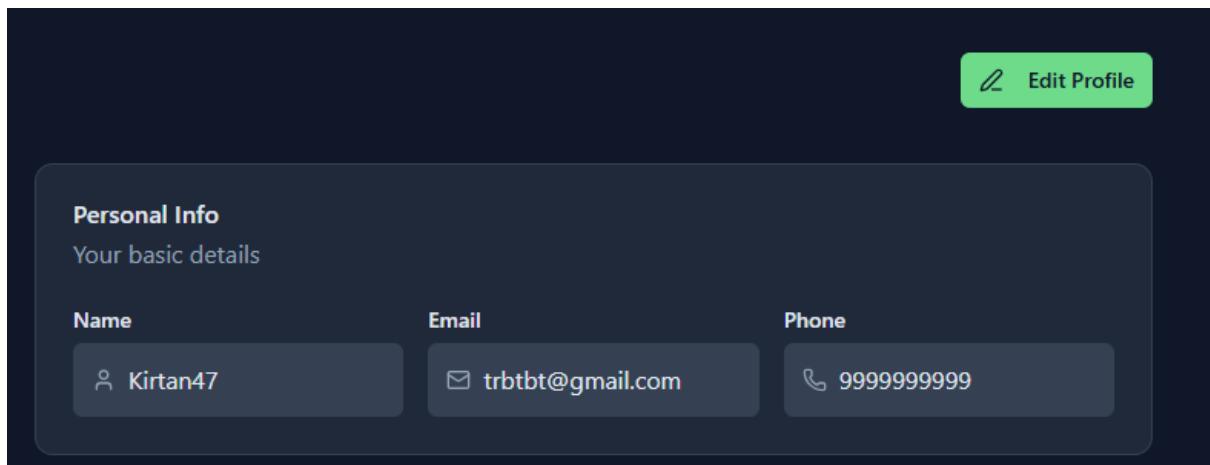
#### **Test 7 — Update Phone Number Successfully**

##### **Action:**

Click **Edit Profile**, set Phone → “9876543210”, click **Save Changes**.

##### **Expected Output:**

Phone number updates and displays on Profile page.



For Create new password :

**Test 1 — Empty Fields (All Blank)**

**Current Password:** (empty)

**New Password:** (empty)

**Confirm Password:** (empty)

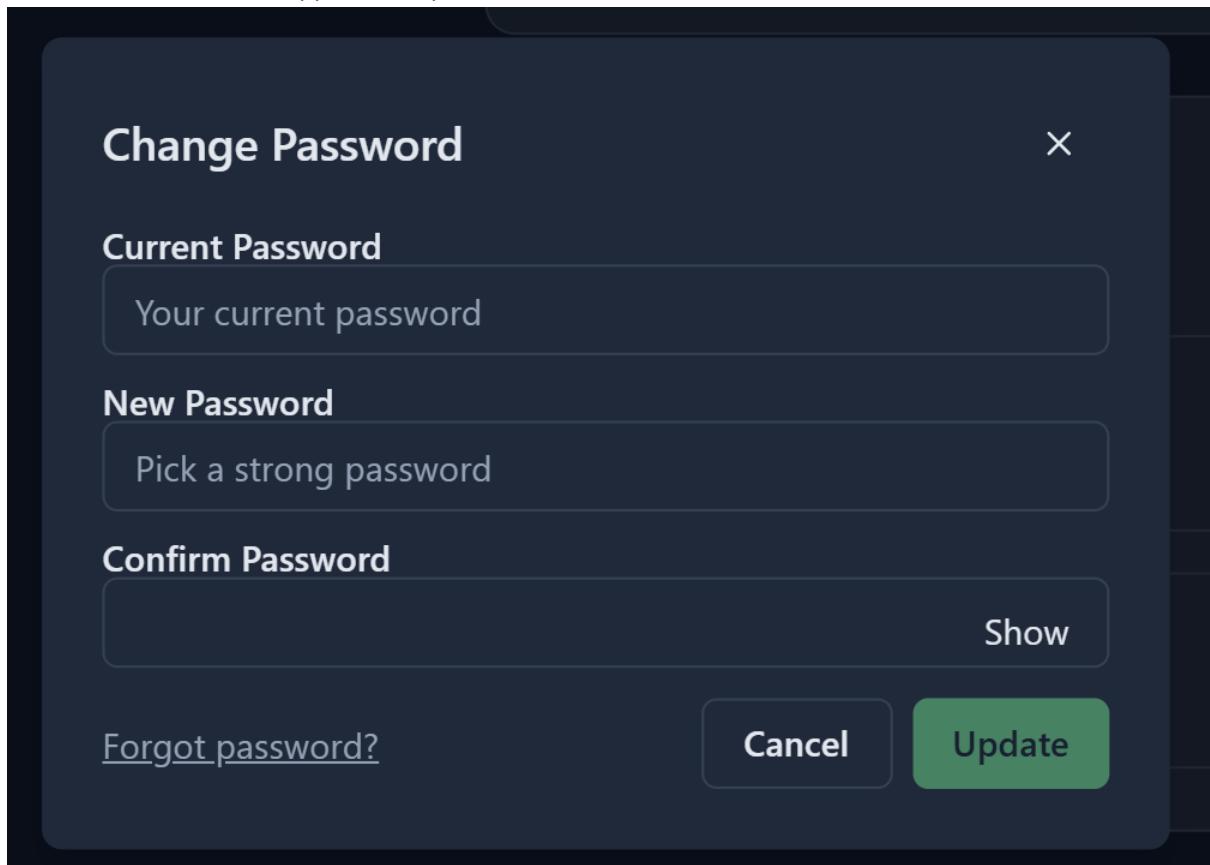
**Action:**

Click **Update**.

**Expected Output:**

Form should **not submit**.

Validation errors should appear OR Update butt



on stays disabled.

---

## Test 2 — Incorrect Current Password

**Current Password:** wrongOldPass1

**New Password:** Abcd@123

**Confirm Password:** Abcd@123

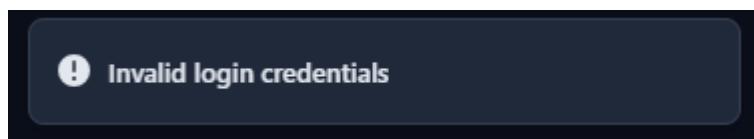
**Action:**

Click **Update**.

**Expected Output:**

Password update fails.

Error message appears: “Incorrect current password” (or your app’s variation).



---

## Test 3 — New Password Too Weak (Less Than 8 Characters)

**Current Password:** correctOldPass

**New Password:** 12345

**Confirm Password:** 12345

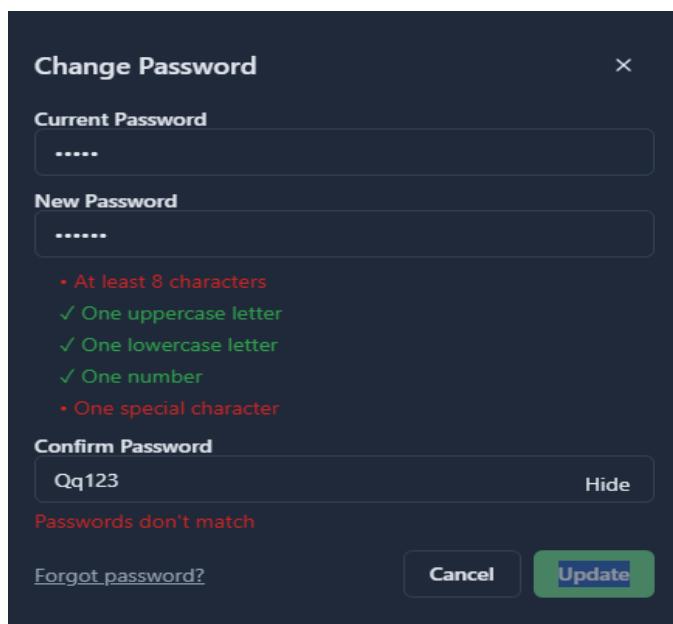
**Action:**

Click **Update**.

**Expected Output:**

Update prevented.

Validation message: “Password must be at least 8 characters” (or similar).



#### Test 4 — New Password and Confirm Password Do Not Match

**Current Password:** correctOldPass

**New Password:** Abcd@123

**Confirm Password:** Abcd@12

**Action:**

Click **Update**.

**Expected Output:**

Update fails.

Error message: “Passwords do not match.”

The screenshot shows a 'Change Password' form. The 'New Password' field contains 'Abcd@123', which is highlighted in red. The 'Confirm Password' field contains 'Abcd@12', which is also highlighted in red. A red error message 'Passwords don't match' is displayed below the fields. The 'Update' button at the bottom right is highlighted with a green background.

#### Test 5 — Using the Same Password as Old Password

**Current Password:** correctOldPass

**New Password:** correctOldPass

**Confirm Password:** correctOldPass

**Action:**

Click **Update**.

**Expected Output:**

Update is rejected.

Error message: “”



---

#### Test 6 — Successful Password Change

**Current Password:** correctOldPass

**New Password:** StrongPass@123

**Confirm Password:** StrongPass@123

**Action:**

Click **Update**.

**Expected Output:**

Password updated successfully.

Modal closes OR success toast appears.

📸 **Screenshot:** (Insert screenshot of success message)

---

#### Test 7 – Forgot password

*Action : Click on Forgot password*

*Expected output :*

