

Quality Assurance Testing Assignment

Name: Mohammad Ali

Date: 15/01/2025

Subject: Gurucool Astrology App Testing

Introduction:

This document outlines my approach to testing the Gurucool astrology app, focusing on manual and automated testing techniques. The assignment includes detailed test cases, a bug report, a proposed test strategy for notifications, and adherence to the evaluation criteria.

1. Objective

This assignment evaluates your ability to perform manual and automated testing, identify critical issues, and document them comprehensively for the Gurucool astrology app.

2. Scenario Overview

Gurucool is a mobile application providing astrology services such as personalized readings, horoscope predictions, and consultations with astrologers. Users can:

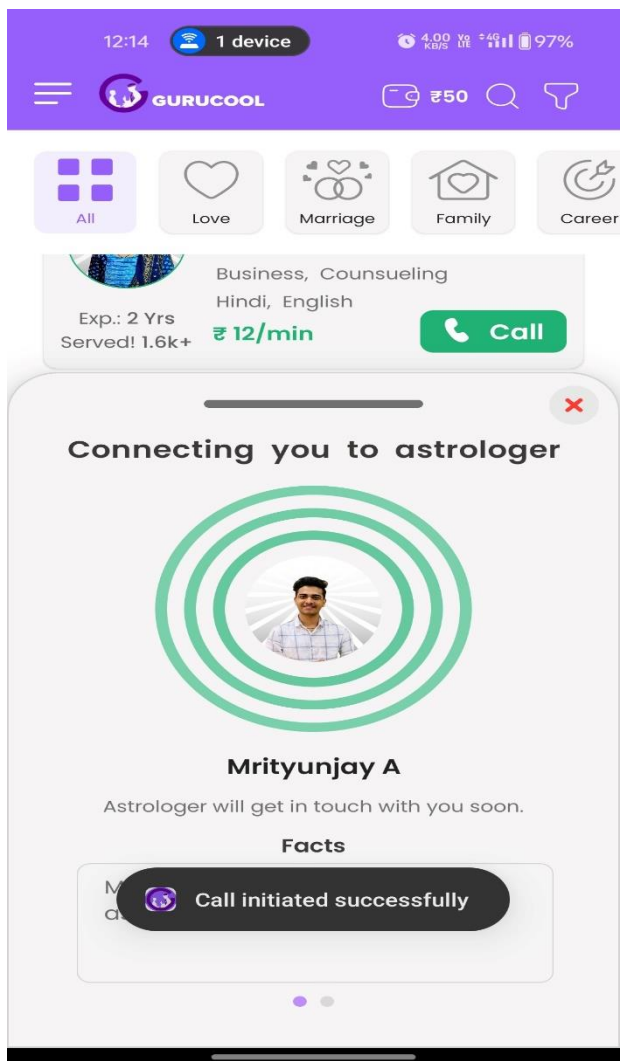
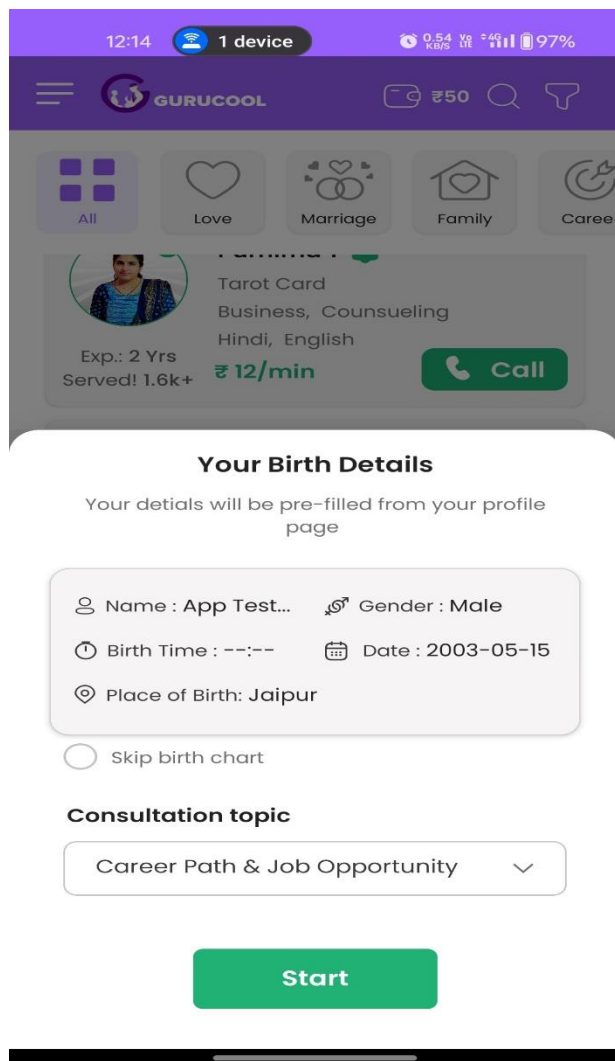
1. Register and log in.
2. Book sessions with astrologers.
3. View daily horoscopes.
4. Manage payments through Razorpay.
5. Provide feedback after consultations.

3. Part 1: Test Case Writing

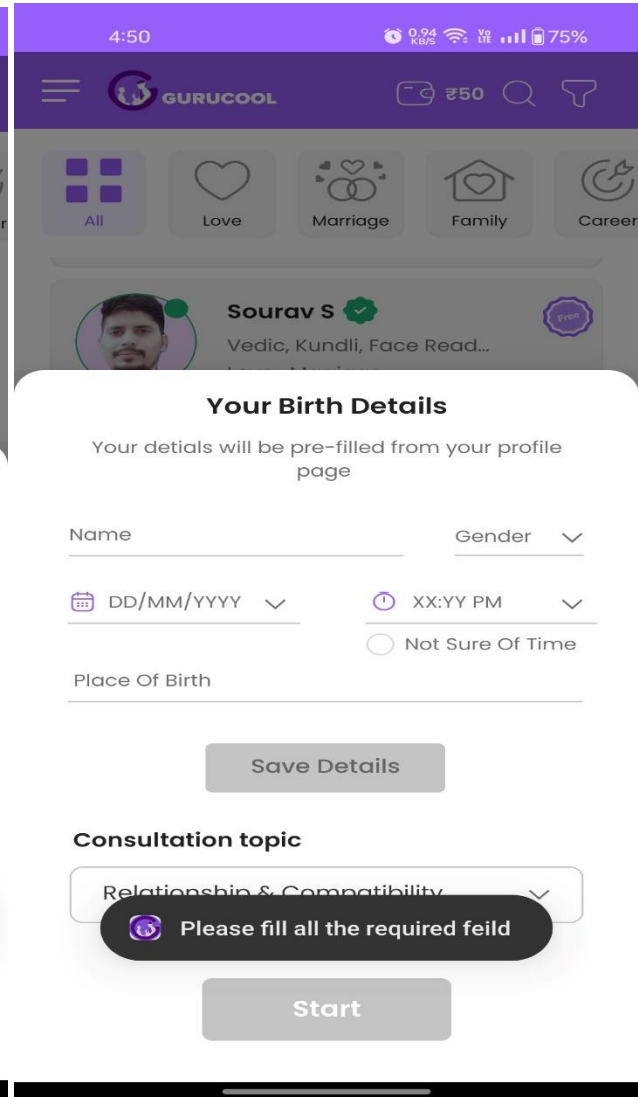
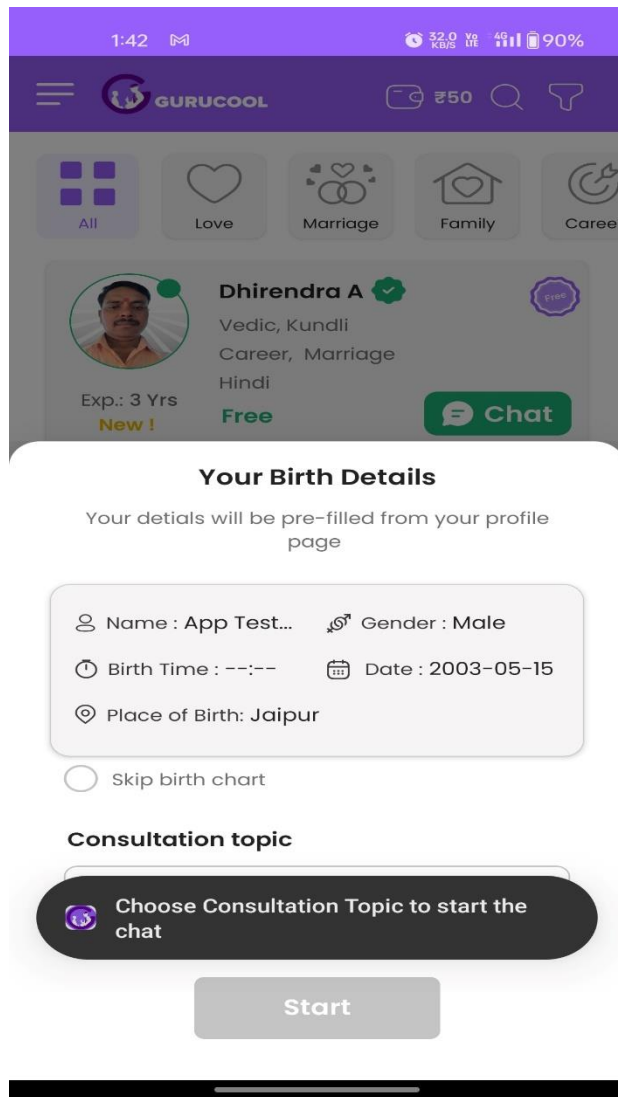
Feature: Booking an Astrology Session

- Write five detailed test cases for the given scenarios:
 - Successful booking with valid inputs.
 - Booking attempt without entering required details.
 - Double booking for the same time slot.
 - Canceling a booked session.
 - Payment processing failure through Razorpay.
- Include these fields for each test case:
 - Test Case ID, Description, Preconditions, Test Steps, Expected Results, Actual Results

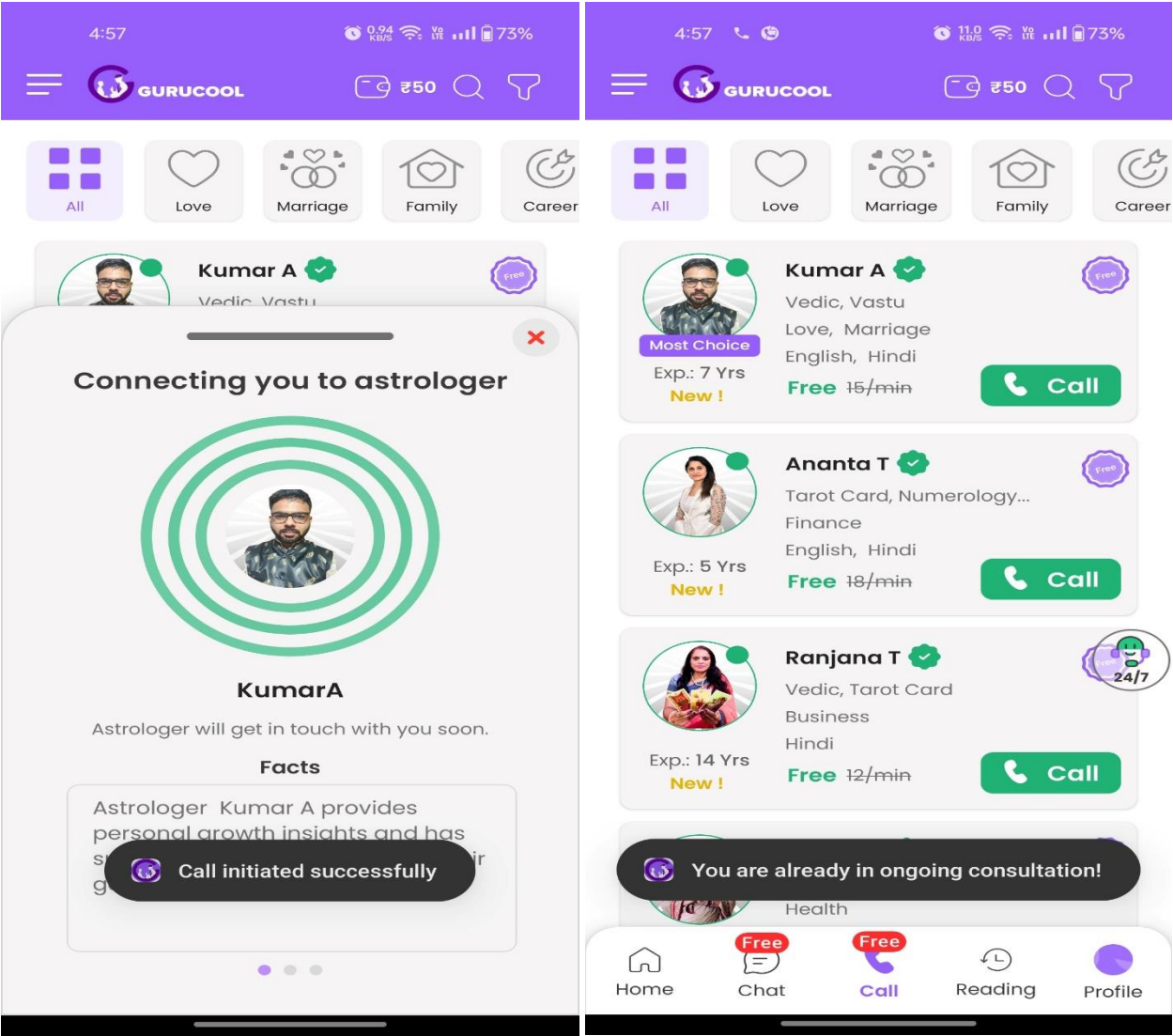
Test Case 1: Successful Booking with Valid Inputs	
Test Case ID	TC_Booking_001
Description	Validate that a user can successfully book an astrology session with valid inputs.
Preconditions	<ul style="list-style-type: none"> User is logged into the Gurucool app. Should have money in the "Wallet". The astrologer has available time slots.
Test Steps	<ol style="list-style-type: none"> 1. Navigate to the "Chat" or "Call" feature. 2. Choose Between available astrologers. 3. Enter required details (e.g., name, topic). 4. Confirm the booking by clicking "Start".
Expected Result	The session is booked successfully, and the user receives a confirmation alert. (e.g., "Call initiated successfully")
Actual Result	The session is booked successfully, and the user receives a confirmation alert. (e.g., "Call initiated successfully")



Test Case 2: Booking Attempt Without Entering Required Details	
Test Case ID	TC_Booking_002
Description	Validate that the app prompts the user to enter required details when missing.
Preconditions	<ul style="list-style-type: none"> User is logged into the Gurucool app. Should have money in the "Wallet". The astrologer has available time slots.
Test Steps	<ol style="list-style-type: none"> Navigate to the "Chat" or "Call" feature. Choose Between available astrologers. Confirm the booking by clicking "Start".
Expected Result	The system displays an error message indicating that all required fields must be filled.
Actual Result	The system displays an error message indicating that all required fields must be filled.

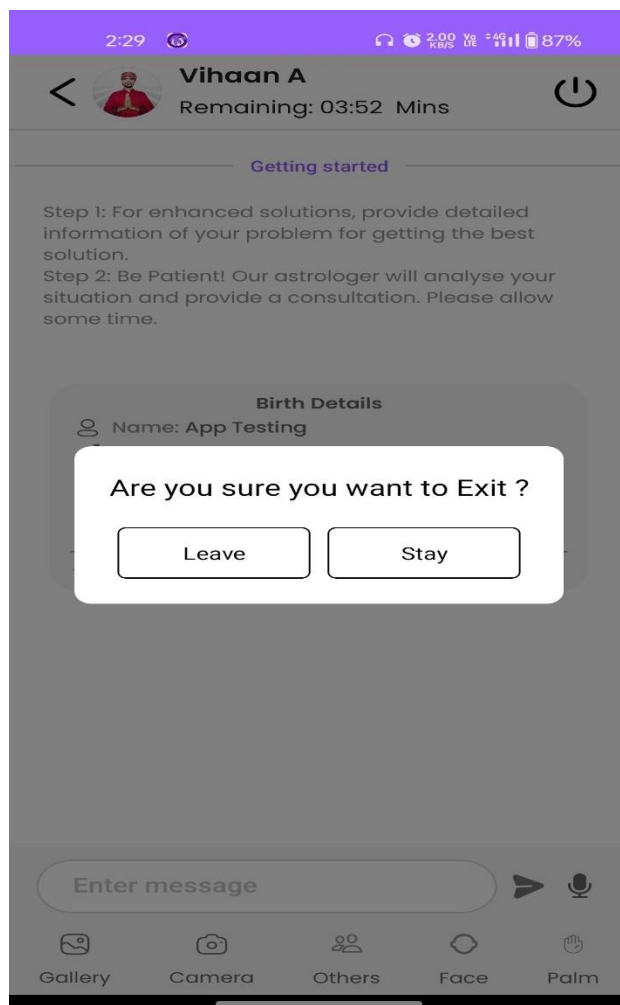


Test Case 3: Double Booking for the Same Time Slot	
Test Case ID	TC_Booking_003
Description	Ensure the app prevents double booking for the same time slot.
Preconditions	<ul style="list-style-type: none">User has already booked a session.
Test Steps	<ol style="list-style-type: none">Attempt to book another session.Confirm the booking.
Expected Result	The system displays a message preventing double booking for the same slot. (e.g., "You are already in ongoing consultation")
Actual Result	The system displays a message preventing double booking for the same slot. (e.g., "You are already in ongoing consultation")

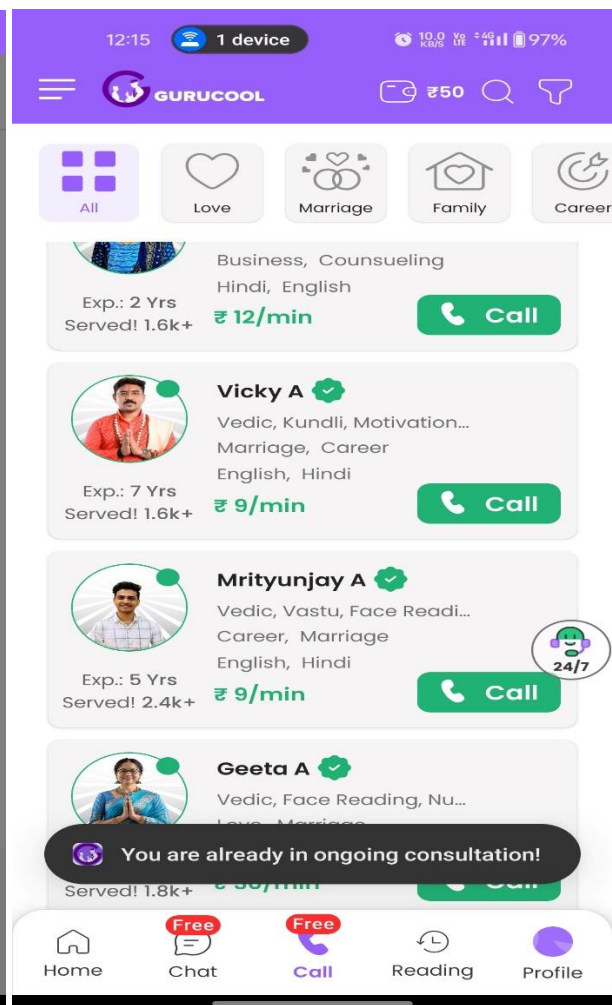


Test Case 4: Canceling a Booked Session	
Test Case ID	TC_Booking_004
Description	Validate that users can cancel a booked session.
Preconditions	<ul style="list-style-type: none"> User has an active booking.
Test Steps	<ol style="list-style-type: none"> 1. Navigate to the "Chat" session. 2. Click on the "Exit" option. 3. Confirm the cancellation by clicking "Leave". 4. Verify that the previously booked slot is now available. 5. Navigate to the "Call" session. 6. Attempt to cancel the session.
Expected Results	<ul style="list-style-type: none"> The session is successfully canceled, and the slot becomes available for other users. Users should have a clear option to cancel sessions for both "Chat" and "Call" features.
Actual Result	The session is canceled for "Chat," but there is no cancel option available for the "Call" feature.

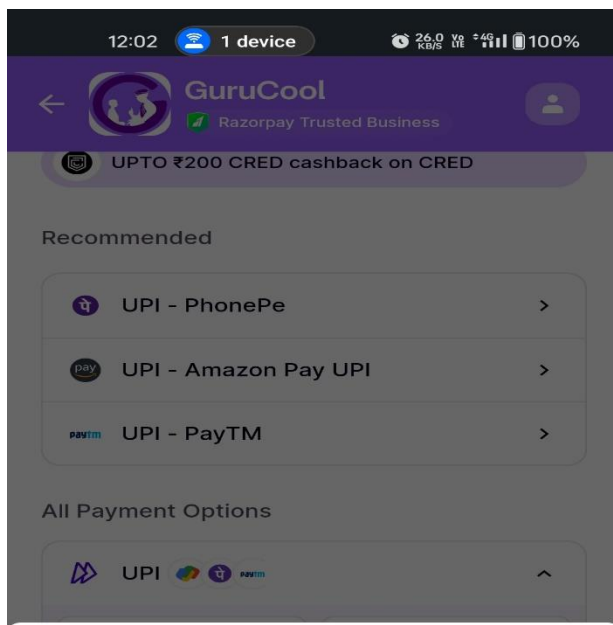
Chat



Call



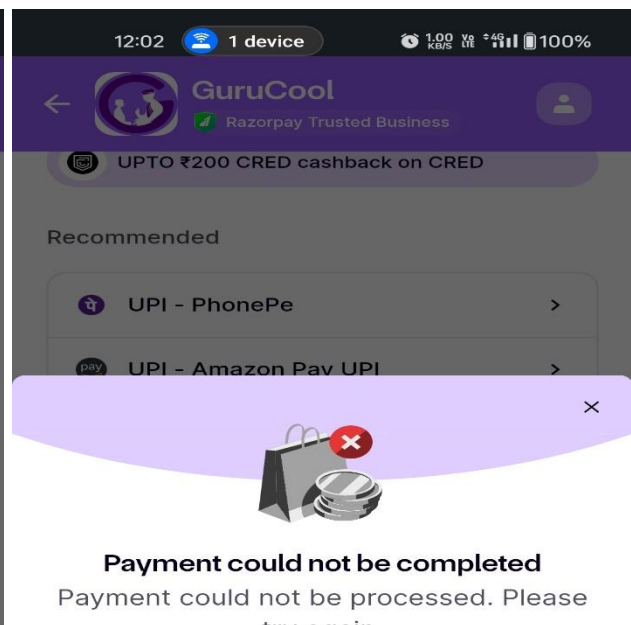
Test Case 5: Payment Processing Failure through Razorpay	
Test Case ID	TC_Booking_005
Description	Verify that the system handles payment failures gracefully.
Preconditions	<ul style="list-style-type: none"> User is logged into the Gurucool app.
Test Steps	<ol style="list-style-type: none"> 1. Navigate to the "Wallet" feature. 2. Attempt to add money using Razorpay. 3. Simulate a payment failure scenario (e.g., insufficient funds, network error). 4. Observe the system's behavior.
Expected Result	The system notifies the user of the payment failure and allows retrying or choosing another payment method.
Actual Result	The system notifies the user of the payment failure and allows retrying or choosing another payment method.



Payment Failed



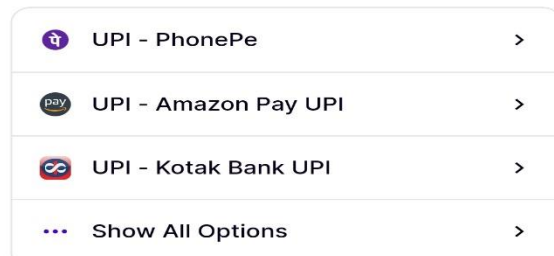
Secured by Razorpay



Payment could not be completed

Payment could not be processed. Please try again.

Retry payment of ₹5,900 with



Secured by Razorpay

4. Part 2: Bug Report

Scenario: App Crash in Horoscope Section

- Include the following details in your bug report:
 - Bug ID
 - Summary
 - Steps to Reproduce
 - Expected vs. Actual Results
 - Severity and Priority
 - Environment Details (e.g., device model, OS version, app version)

Bug Report		
Bug ID	BUG_Horoscope_001	
Summary	The app crashes unexpectedly when accessing the "Lucky Number" section for Virgo in the horoscope feature.	
Steps to Reproduce	<ol style="list-style-type: none">1. Launch the Gurucool app.2. Log in with valid credentials.3. Navigate to the "Horoscope" section.4. Select the zodiac sign "Virgo."5. Gets the "Lucky Number" for Virgo.	
Expected Result	The "Lucky Number" for Virgo should be displayed successfully without any crashes.	
Actual Result	The app crashes unexpectedly when accessing the "Lucky Number" section for Virgo.	
Severity: High	The crash impacts the user experience for horoscope users, specifically Virgo, and makes the feature unusable.	
Priority: Critical	Immediate resolution is required as this is a core functionality of the app.	
Environment Details	Device Model	OPPO F21 PRO
	OS Name	ColorOS
	OS Version	Android 14
	App Version	0.1.41
	Screen Resolution	1800 x 2400 pixels

Note: While my primary assignment was testing the Gurucool app, I also wrote a Selenium script to perform the same operation on the website. This showcases my proficiency in Selenium for automated testing.


```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time

# Initialize the Edge driver
driver = webdriver.Edge()
url = "https://www.gurucool.life/horoscope"
driver.get(url)
driver.maximize_window()

# Click on the login button
login = driver.find_element(By.XPATH, "//span[@class='select-none']")
login.click()

# Enter phone number and get OTP
number = "1234567890"
number_field = driver.find_element(By.XPATH,
    "//input[@placeholder='Enter phone number']")
number_field.send_keys(number)
get_otp = driver.find_element(By.XPATH, "//button[normalize-
space()='GET OTP']")
get_otp.click()

# Wait for user to enter OTP manually and click submit
time.sleep(30)

# Click on the dropdown to for getting Virgo
dropdown_element = driver.find_element(By.CSS_SELECTOR, "body >
main:nth-child(5) > div:nth-child(1) > div:nth-child(3) > div:nth-
child(1) > div:nth-child(1) > button:nth-child(1)")
dropdown_element.click()

# Click on the Virgo button
select_virgo = driver.find_element(By.CSS_SELECTOR, "body > main:nth-
child(5) > div:nth-child(1) > div:nth-child(3) > div:nth-child(1) >
div:nth-child(1) > div:nth-child(2) > div:nth-child(6)")
select_virgo.click()

# Check for the Virgo heading
virgo = driver.find_element(By.XPATH, "//h2[normalize-
space()='Virgo']").text
try:
    assert virgo == "Virgo"
    print("Virgo Lucky Number accessed!")
except AssertionError:
    print("Virgo Lucky Number not accessed!")

# Wait before closing the driver
time.sleep(10)
driver.quit()
```


5. Bonus Task: Test Strategy for Daily Horoscope Notifications

Objective

To ensure that users receive daily horoscope notifications on time and without duplication across all supported devices and platforms.

Test Plan		
Test Scenarios	Timely Notification Delivery	<ul style="list-style-type: none">Validate notifications are sent at the scheduled time (e.g., 8:00 AM daily).Test across different time zones to ensure correct delivery time.
	No Duplication	<ul style="list-style-type: none">Ensure no user receives duplicate notifications on the same day.
	Notification Content	<ul style="list-style-type: none">Verify the content matches the horoscope for the day and the user's zodiac sign.Ensure the content is correctly localized (if multiple languages are supported).
	Opt-In/Out Behavior	<ul style="list-style-type: none">Verify notifications are sent only to users who have opted in.Test the toggle to enable/disable notifications and validate behavior.
	Device and OS Compatibility	<ul style="list-style-type: none">Test notification delivery across various devices and OS versions (Android, iOS).
	Network Conditions	<ul style="list-style-type: none">Test notification delivery under varying network conditions. (e.g., Wi-Fi, Mobile Data, No Internet)
	Failure Scenarios	<ul style="list-style-type: none">Simulate server downtime to ensure notifications are queued and sent later.Test app behavior when a user changes the device or reinstalls the app.
Test Cases	<ol style="list-style-type: none">Validate notification is sent at the scheduled time.Ensure users do not receive duplicate notifications.Verify correct zodiac sign and horoscope content in the notification.Check notifications are suppressed for users who have opted out.Confirm notifications are displayed correctly on locked and unlocked screens.Test retry mechanism for failed notifications due to poor connectivity.	
Testing Tools	<ul style="list-style-type: none">Analyze logs for sent, delivered, and failed notifications.	
Metrics to Track	<ul style="list-style-type: none">Delivery success rate (% of notifications sent successfully).Duplicate notification rate (should be 0%).Latency (time taken for notification to reach users).	
Regression Testing	<ul style="list-style-type: none">After every update to the notification system, perform regression testing to ensure previous functionality is not broken.	
Automation Opportunities	<ul style="list-style-type: none">Automate tests for notification delivery timing, duplication, and content validation using tools like (Appium, Selenium) or Postman for API testing.	