

BOOKINGS MODULE – TEST CASE REPORT

A. Bookings – Unit & Helper Test Cases

Table A1 – Booking & Payment model + helper functions

Sr. No.	Test Case ID	Test Name / Area	Brief Scenario	Expected Result	Actual Result
1	TC_BKG_MO DEL_01	Booking model – string & reference	Create a Booking with valid Client and Counsellor, then save without manually setting booking_reference.	booking_reference auto-generated in format like MBK-XXXXXXXXXX, and str(booking) shows reference + client → counsellor names.	As Expected (Pass)
2	TC_BKG_MO DEL_02	Booking model – session_datetime	Set session_date and session_time on a Booking.	session_datetime combines date + time into a timezone-aware datetime.	As Expected (Pass)
3	TC_PAY_MOD EL_01	Payment model – string & ID generation	Create Payment linked to a Booking without setting payment_id.	On save, payment_id auto-generated in format like PAY- XXXXXXXXXXXX, and str(payment) shows human-readable status.	As Expected (Pass)
4	TC_PAY_MOD EL_02	Payment model – state transitions	Call mark_success(...) and mark_failed(...) on a Payment.	mark_success sets status=success, stores Razorpay IDs, sets paid_at; mark_failed sets status=failed and stores error message/payload.	As Expected (Pass)

5	TC_HELPER_01	_parse_time_slot helper	Call _parse_time_slot with valid string (e.g. "10:30") and invalid string (e.g. "25:99").	Valid input returns a time(10, 30) object; invalid input returns None without raising an exception.	As Expected (Pass)
6	TC_HELPER_02	_validate_session_date helper	Call _validate_session_date with past date, invalid format, and valid future date.	Past/invalid dates return None; valid future date returns a parsed date object that can be used for booking.	As Expected (Pass)

B. Bookings – Integration Test Cases

All of these use Django's test client and hit the **bookings URLs**:

- POST /bookings/create/
- POST /bookings/verify/
- POST /bookings/payment-failed/

Table B1 – Create booking endpoint (bookings:create_booking)

Sr. No.	Test Case ID	Test Function Name	Brief Scenario	Expected Result	Actual Result
7	TC_BKG_01	test_create_booking_invalid_payload	Logged-in client sends a malformed / non-JSON body to /bookings/create/.	returns 400 Bad Request with success=False and error "Invalid payload ."	As Expected (Pass)

8	TC_BKG_02	test_create_booking_requires_client_role	Logged-in user who is not a Client tries to create a booking.	returns 403 Forbidden with success=False and message like "Only clients can book sessions."	As Expected (Pass)
9	TC_BKG_03	test_create_booking_enforces_minimum_days	Logged-in client tries to book a slot with session_date less than 3 days from today.	rejects request with 400 and error "Appointments must be booked at least 3 days in advance."	As Expected (Pass)
10	TC_BKG_04	test_create_booking_success	Happy path: client, counsellor, valid future date (≥ 3 days), available CounsellorAvailability slot; Razorpay client mocked.	returns 200 OK with success=True. Booking and Payment records are created, slot is marked booked, and response includes booking reference, date, time, fee, and Razorpay order details.	As Expected (Pass)

**Table B2 – Payment verification & failure
(bookings:verify_payment, bookings:payment_failed)**

Sr. No.	Test Case ID	Test Function Name	Brief Scenario	Expected Result	Actual Result
11	TC_BKG_05	test_verify_payment_invalidate_signature	Logged-in client sends verify payload where Razorpay signature check is mocked to raise SignatureVerificationError.	returns 400 with success=False. Payment status becomes failed, and linked Booking.payment_status becomes failed.	As Expected (Pass)

12	TC_BKG_06	test_verify_payment_success	Valid verification: Razorpay verify_payment_signature mocked to succeed for a pending payment + booking.	returns 200 with success=True. Payment status = success; Booking.status = confirmed, Booking.payment_status = paid. Client.total_sessions increments and last_session_date is set; Counsellor.total_sessions (and total_clients for first paid session) update correctly.	As Expected (Pass)
13	TC_BKG_07	test_payment_failed_marks_failed_and_releases_slot	Razorpay checkout failure is posted to /bookings/payment-failed/ with error details and booking reference.	Payment status becomes failed, Booking.payment_status = failed, and associated availability slot is unbooked (is_booked=False) so it can be reused. returns 200 with success=True.	As Expected (Pass)

Bookings module testing summary:

A total of **13 automated tests** (6 unit/helper tests and 7 integration tests) were executed for the Bookings module, covering core model logic, helper utilities, and end-to-end booking and payment flows. **All 13/13 tests passed successfully**, ensuring that booking creation, Razorpay payment verification, error handling, and slot management behave as expected under both normal and edge conditions.

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The title bar reads "Mind_Ease_V". The left sidebar (Explorer) displays a file tree for a project named "MIND_EASE_V". The "tests.py" file is currently selected. The main area (Terminal) shows the output of a command-line session:

```
(.venv) C:\Users\Himani\OneDrive\Desktop\Mind_Ease_V>python manage.py test bookings
C:\Users\Himani\OneDrive\Desktop\Mind_Ease_V\.venv\Lib\site-packages\razorpay\client.py:4: UserWarning: pkg_resources is deprecated as an API.
See https://setuptools.pypa.io/en/latest/pkg_resources.html. The pkg_resources package is slated for removal as early as 2025-11-30. Refrain from using this package or pin to Setuptools<81.
    import pkg_resources
    Found 13 test(s).
    Creating test database for alias 'default'...
    System check identified no issues (0 silenced).
.....Razorpay order created: order_123 for booking MBK-E4DB1BD5FF, amount: 90000 paise
.Payment failed for booking MBK-51F6566E9A:
    Description: Payment failed at bank
    Code: BAD_REQ
    Source: bank
    Step: payment_auth
    Reason: declined_by_bank
    Full error: {'description': 'Payment failed at bank', 'code': 'BAD_REQ', 'source': 'bank', 'step': 'payment_auth', 'reason': 'declined_by_bank'}
.....
-----
Ran 13 tests in 16.042s

OK
Destroying test database for alias 'default'...

(.venv) C:\Users\Himani\OneDrive\Desktop\Mind_Ease_V>
```

The status bar at the bottom indicates the terminal has 320 columns and 36 rows, with 4 spaces, using UTF-8 encoding, and is connected to Python 3.11.4. There is also a "Go Live" button.