**✅ Testing Approach**

**🎯 Objectives**

* Validate core business logic (e.g. capacity checks, duplicate bookings)
* Ensure correct HTTP responses and data formats
* Verify route protection (auth vs public)
* Catch regressions through automated testing

**🧪 Types of Tests**

**1. Unit Tests**

Test isolated logic (e.g. model methods, services)

✅ Examples:

* BookingTest:
  + Ensures an attendee cannot be booked twice for the same event
  + Fails booking if event is at capacity
* EventTest:
  + Validates time logic (e.g. start\_time < end\_time)

📁 Location: tests/Unit/

**2. Feature Tests**

Simulate real HTTP requests and validate API behavior

✅ Examples:

* EventApiTest:
  + Test POST /events with/without auth
  + Test validation errors (missing name, invalid time)
* AttendeeApiTest:
  + Register attendee (valid + duplicate email case)
* BookingApiTest:
  + Book event successfully
  + Attempt to overbook event (capacity hit)
  + Attempt duplicate booking

📁 Location: tests/Feature/

**🔒 Security Tests**

* Unauthorized access to protected endpoints (e.g. booking list, event update)
* Authenticated but unauthorized user attempting to update another manager's event

**📦 Validation Tests**

* Event creation: invalid country, missing name, bad date format
* Booking: invalid event\_id, non-existent attendee\_id

Use assertJsonValidationErrors() in Laravel Feature tests to confirm these.