

# Test Case Report (BrandBoost)

# Introduction

This report organize the test cases by using each controller. Each table includes the Test Case ID, Description, Precondition, Test Steps, Expected Result, and Test Type, as previously outlined. Below are the test cases, with one table per test case, grouped by controller.

## 1. CustomPackageController

Test Case ID	CPC-01
Description	Get custom package list by seller
Precondition	User is logged in, valid seller_id exists in the database
Test Steps	1. Log in as a user with valid credentials. 2. Send a GET request to <code>/custom-packages?seller=true&amp;seller_id=1</code> .
Expected Result	Returns a JSON response containing a list of custom packages associated with seller_id=1.
Test Type	Positive

Test Case ID	CPC-02
Description	Get custom package list by buyer
Precondition	User is logged in, valid buyer_id exists in the database
Test Steps	1. Log in as a user with valid credentials. 2. Send a GET request to <code>/custom-packages?buyer=true&amp;buyer_id=1</code> .
Expected Result	Returns a JSON response containing a list of custom packages associated with buyer_id=1.
Test Type	Positive

Test Case ID	CPC-03
Description	Get custom package list without parameters
Precondition	User is logged in
Test Steps	1. Log in as a user with valid credentials. 2. Send a GET request to /custom-packages.
Expected Result	Returns a JSON response containing all custom packages accessible to the user.
Test Type	Positive

Test Case ID	CPC-04
Description	Get custom package list with invalid seller_id
Precondition	User is logged in
Test Steps	1. Log in as a user with valid credentials. 2. Send a GET request to /custom-packages?seller=true&seller_id=invalid.
Expected Result	Returns a JSON response with an error message (e.g., "Invalid Input") or empty list.
Test Type	Negative

Test Case ID	CPC-05
Description	Get custom package profile with valid ID
Precondition	Valid custom_package_id exists in the database
Test Steps	1. Send a GET request to /custom-package/1.
Expected Result	Returns a JSON response with custom package details, including associated service data.

Test Type	Positive
-----------	----------

Test Case ID	CPC-06
Description	Get custom package profile with non-existent ID
Precondition	None
Test Steps	1. Send a GET request to /custom-package/999.
Expected Result	Returns a JSON response with an empty result or error message (e.g., "Not found").
Test Type	Negative

Test Case ID	CPC-07
Description	Create custom package with valid data
Precondition	User is logged in, valid service_id exists
Test Steps	1. Log in as a user with valid credentials. 2. Send a POST request to /custom-package with JSON payload: {"service_id": 1, "benefits": "Test", "delivery": 5, "revisions": 2, "price": 100}.
Expected Result	Returns a JSON response with {"success": true}.
Test Type	Positive

Test Case ID	CPC-08
Description	Create custom package with invalid service_id
Precondition	User is logged in

Test Steps	<ol style="list-style-type: none"> <li>1. Log in as a user with valid credentials.</li> <li>2. Send a POST request to /custom-package with JSON payload: {"service_id": 999}.</li> </ol>
Expected Result	Returns a JSON response with {"success": false}.
Test Type	Negative

Test Case ID	CPC-09
Description	Update custom package with valid data
Precondition	User is logged in, valid custom_package_id exists
Test Steps	<ol style="list-style-type: none"> <li>1. Log in as a user with valid credentials.</li> <li>2. Send a POST request to /custom-package/1 with JSON payload: {"benefits": "Updated", "delivery": 7, "revisions": 3, "price": 150}.</li> </ol>
Expected Result	Returns a JSON response with {"success": true}.
Test Type	Positive

Test Case ID	CPC-10
Description	Update custom package with non-existent ID
Precondition	User is logged in
Test Steps	<ol style="list-style-type: none"> <li>1. Log in as a user with valid credentials.</li> <li>2. Send a POST request to /custom-package/999 with JSON payload: {"benefits": "Updated"}.</li> </ol>
Expected Result	Returns a JSON response with {"success": false}.
Test Type	Negative

Test Case ID	CPC-11
Description	Delete custom package with valid ID
Precondition	User is logged in, valid custom_package_id exists
Test Steps	1. Log in as a user with valid credentials. 2. Send a DELETE request to /custom-package/1.
Expected Result	Returns a JSON response with {"success": true}.
Test Type	Positive

Test Case ID	CPC-12
Description	Delete custom package with non-existent ID
Precondition	User is logged in
Test Steps	1. Log in as a user with valid credentials. 2. Send a DELETE request to /custom-package/999.
Expected Result	Returns a JSON response with {"success": false}.
Test Type	Negative

Test Case ID	CPC-13
Description	Create custom package without authentication
Precondition	User is not logged in
Test Steps	1. Send a POST request to /custom-package with valid JSON payload.
Expected Result	Returns a JSON response with an unauthorized error (e.g., 401).
Test Type	Negative

Test Case ID	CPC-14
Description	Get custom package list with both seller and buyer parameters
Precondition	User is logged in
Test Steps	<ol style="list-style-type: none"> <li>1. Log in as a user with valid credentials.</li> <li>2. Send a GET request to /custom-packages?seller=true&amp;seller_id=1&amp;buyer=true&amp;buyer_id=2.</li> </ol>
Expected Result	Returns an error JSON or prioritizes one filter (based on implementation logic).
Test Type	Edge

## 2. ComplaintController

Test Case ID	CC-01
Description	Get complaint list as admin
Precondition	User is logged in as admin
Test Steps	<ol style="list-style-type: none"> <li>1. Log in as an admin user.</li> <li>2. Send a GET request to /complaints?limit=10&amp;offset=0.</li> </ol>
Expected Result	Returns a JSON response with a paginated list of complaints and pagination metadata.
Test Type	Positive

Test Case ID	CC-02
Description	Get complaint list as non-admin
Precondition	User is logged in as non-admin
Test Steps	<ol style="list-style-type: none"> <li>1. Log in as a non-admin user.</li> <li>2. Send a GET request to /complaints.</li> </ol>

Expected Result	Returns a JSON response with {"error": "Unauthorized"} and status code 403.
Test Type	Negative

Test Case ID	CC-03
Description	Get complaint list with filters
Precondition	User is logged in as admin, complaints exist
Test Steps	<ol style="list-style-type: none"> <li>1. Log in as an admin user.</li> <li>2. Send a GET request to /complaints?status=open,pending&amp;date_from=2025-01-01.</li> </ol>
Expected Result	Returns a JSON response with complaints filtered by status and date.
Test Type	Positive

Test Case ID	CC-04
Description	Get complaint list with invalid status
Precondition	User is logged in as admin
Test Steps	<ol style="list-style-type: none"> <li>1. Log in as an admin user.</li> <li>2. Send a GET request to /complaints?status=invalid.</li> </ol>
Expected Result	Returns a JSON response with an error or empty list.
Test Type	Negative

Test Case ID	CC-05
Description	Get complaint details as admin
Precondition	User is logged in as admin, valid complaint_id exists
Test Steps	<ol style="list-style-type: none"> <li>1. Log in as an admin user.</li> <li>2. Send a GET request to /complaint/1.</li> </ol>



Expected Result	Returns a JSON response with detailed complaint data, including order and order promises.
Test Type	Positive

Test Case ID	CC-06
Description	Get complaint details with invalid ID
Precondition	User is logged in as admin
Test Steps	1. Log in as an admin user. 2. Send a GET request to /complaint/999.
Expected Result	Returns a JSON response with {"error": "Complaint not found"} and status code 404.
Test Type	Negative

Test Case ID	CC-07
Description	Create complaint with valid data
Precondition	User is logged in, valid order_id exists
Test Steps	1. Log in as a user. 2. Send a POST request to /complaint with form-data: order_id=1, content="Issue", complaint_type="delay", proofs=[valid_file].
Expected Result	Returns a JSON response with {"success": true, "message": "Complaint submitted successfully"}.
Test Type	Positive

Test Case ID	CC-08
Description	Create complaint with missing fields
Precondition	User is logged in
Test Steps	1. Log in as a user. 2. Send a POST request to /complaint without order_id.

Expected Result	Returns a JSON response with {"success": false, "message": "No data provided for complaint"}.
Test Type	Negative

Test Case ID	CC-09
Description	Create complaint with invalid file upload
Precondition	User is logged in, valid order_id exists
Test Steps	1. Log in as a user. 2. Send a POST request to /complaint with form-data: order_id=1, content="Issue", proofs=[invalid_file].
Expected Result	Returns a JSON response with {"success": false} or partial success with file upload failure.
Test Type	Negative

Test Case ID	CC-10
Description	Update complaint status as admin
Precondition	User is logged in as admin, valid complaint_id exists
Test Steps	1. Log in as an admin user. 2. Send a POST request to /complaint/status with JSON: {"complaint_id": 1, "status": "resolved"}.
Expected Result	Returns a JSON response with {"success": true, "message": "Complaint status updated successfully"}.
Test Type	Positive

Test Case ID	CC-11
Description	Update complaint status as non-admin
Precondition	User is logged in as non-admin
Test Steps	1. Log in as a non-admin user. 2. Send a POST request to /complaint/status.

Expected Result	Returns a JSON response with {"error": "Unauthorized"} and status code 403.
Test Type	Negative

Test Case ID	CC-12
Description	Update complaint status with invalid status
Precondition	User is logged in as admin
Test Steps	<ol style="list-style-type: none"> <li>1. Log in as an admin user.</li> <li>2. Send a POST request to /complaint/status with JSON: {"complaint_id": 1, "status": "invalid"}.</li> </ol>
Expected Result	Returns a JSON response with {"error": "Invalid status value"} and status code 400.
Test Type	Negative

Test Case ID	CC-13
Description	Create complaint with multiple file uploads
Precondition	User is logged in, valid order_id exists
Test Steps	<ol style="list-style-type: none"> <li>1. Log in as a user.</li> <li>2. Send a POST request to /complaint with form-data: order_id=1, content="Issue", proofs=[file1, file2].</li> </ol>
Expected Result	Returns a JSON response with {"success": true}, proofs stored as JSON array in database.
Test Type	Positive

Test Case ID	CC-14
Description	Get complaint list with large limit
Precondition	User is logged in as admin
Test Steps	<ol style="list-style-type: none"> <li>1. Log in as an admin user.</li> <li>2. Send a GET request to /complaints?limit=1000.</li> </ol>

Expected Result	Returns a JSON response with a large dataset or server handles gracefully without crashing.
Test Type	Edge

### 3. GetCountController Test Cases

Test Case ID	GCC-01
Description	Get order counts as admin
Precondition	User is logged in as admin
Test Steps	<ol style="list-style-type: none"> <li>1. Log in as an admin user.</li> <li>2. Send a GET request to /order-counts.</li> </ol>
Expected Result	Returns a JSON response with counts: total, completed, pending, in_progress, canceled.
Test Type	Positive

Test Case ID	GCC-02
Description	Get order counts as seller
Precondition	User is logged in as non-admin (seller)
Test Steps	<ol style="list-style-type: none"> <li>1. Log in as a seller user.</li> <li>2. Send a GET request to /order-counts.</li> </ol>
Expected Result	Returns a JSON response with counts filtered by seller_id.
Test Type	Positive

Test Case ID	GCC-03
Description	Get order counts without login
Precondition	User is not logged in
Test Steps	1. Send a GET request to /order-counts.
Expected Result	Returns a JSON response with {"error": "Unauthorized"} and status code 401.
Test Type	Negative

Test Case ID	GCC-04
Description	Get service counts
Precondition	User is logged in
Test Steps	1. Log in as a user. 2. Send a GET request to /service-counts.
Expected Result	Returns a JSON response with counts: total, gig, promotion.
Test Type	Positive

Test Case ID	GCC-05
Description	Get user counts as admin
Precondition	User is logged in as admin
Test Steps	1. Log in as an admin user. 2. Send a GET request to /user-counts.
Expected Result	Returns a JSON response with counts: total, businessmen, influencers, designers, newSignupsToday.
Test Type	Positive

Test Case ID	GCC-06
Description	Get user counts as non-admin
Precondition	User is logged in as non-admin
Test Steps	1. Log in as a non-admin user. 2. Send a GET request to /user-counts.
Expected Result	Returns a JSON response with {"error": "Unauthorized"} and status code 401.
Test Type	Negative

Test Case ID	GCC-07
Description	Get user counts with sinceTime
Precondition	User is logged in as admin
Test Steps	1. Log in as an admin user. 2. Send a GET request to /user-counts?sinceTime=2025-01-01.
Expected Result	Returns a JSON response with newSignupsToday filtered by sinceTime.
Test Type	Positive

Test Case ID	GCC-08
Description	Get complaint counts as admin
Precondition	User is logged in as admin
Test Steps	1. Log in as an admin user. 2. Send a GET request to /complaint-counts.
Expected Result	Returns a JSON response with counts: total, open, pending, resolved.
Test Type	Positive

Test Case ID	GCC-09
Description	Get complaint counts as non-admin
Precondition	User is logged in as non-admin
Test Steps	1. Log in as a non-admin user. 2. Send a GET request to /complaint-counts.
Expected Result	Returns a JSON response with {"error": "Unauthorized"} and status code 401.
Test Type	Negative

Test Case ID	GCC-10
Description	Get counts with non-GET method
Precondition	User is logged in
Test Steps	1. Log in as a user. 2. Send a POST request to /order-counts.
Expected Result	Returns a JSON response with {"error": "Method Not Allowed"} and status code 405.
Test Type	Negative

## 4. AdminController Test Cases

Test Case ID	AC-01
Description	Access dashboard as admin
Precondition	User is logged in as admin
Test Steps	1. Log in as an admin user. 2. Send a GET request to /admin/dashboard.
Expected Result	Renders the admin dashboard view (pages/admin/dashboard).
Test Type	Positive

Test Case ID	AC-02
Description	Access dashboard as non-admin
Precondition	User is logged in as non-admin
Test Steps	1. Log in as a non-admin user. 2. Send a GET request to /admin/dashboard.
Expected Result	Redirects to an unauthorized page or returns an error.
Test Type	Negative

Test Case ID	AC-03
Description	View services list
Precondition	User is logged in as admin
Test Steps	1. Log in as an admin user. 2. Send a GET request to /admin/services.
Expected Result	Renders the services list view (pages/guest/services_list).
Test Type	Positive

Test Case ID	AC-04
Description	View user profile with valid ID
Precondition	User is logged in as admin, valid user_id exists
Test Steps	1. Log in as an admin user. 2. Send a GET request to /admin/user/1.
Expected Result	Renders the user profile view with user and service data.
Test Type	Positive



Test Case ID	AC-05
Description	View user profile with invalid ID
Precondition	User is logged in as admin
Test Steps	1. Log in as an admin user. 2. Send a GET request to /admin/user/999.
Expected Result	Renders the user profile view with empty or error data.
Test Type	Negative

Test Case ID	AC-06
Description	Access verifications list
Precondition	User is logged in as admin
Test Steps	1. Log in as an admin user. 2. Send a GET request to /admin/verifications.
Expected Result	Renders the verifications list view (pages/admin/verifications_list).
Test Type	Positive

Test Case ID	AC-07
Description	Access complaint details
Precondition	User is logged in as admin
Test Steps	1. Log in as an admin user. 2. Send a GET request to /admin/complaint/1.
Expected Result	Renders the complaint details view (pages/admin/complaint_details).
Test Type	Positive

## 5. BusinessmanController

Test Case ID	BC-01
Description	Access orders list as businessman
Precondition	User is logged in as businessman
Test Steps	1. Log in as a businessman user. 2. Send a GET request to /businessman/orders.
Expected Result	Renders the orders list view (pages/businessman/orders_list).
Test Type	Positive

Test Case ID	BC-02
Description	Access orders list as non-businessman
Precondition	User is logged in as non-businessman
Test Steps	1. Log in as a non-businessman user. 2. Send a GET request to /businessman/orders.
Expected Result	Redirects to an unauthorized page or returns an error.
Test Type	Negative

Test Case ID	BC-03
Description	Request custom package
Precondition	User is logged in as businessman
Test Steps	1. Log in as a businessman user. 2. Send a GET request to /businessman/request-package.
Expected Result	Renders the request package view (pages/businessman/request_package).
Test Type	Positive

Test Case ID	BC-04
Description	View requested custom packages
Precondition	User is logged in as businessman
Test Steps	1. Log in as a businessman user. 2. Send a GET request to /businessman/requested-custom-packages.
Expected Result	Renders the requested custom packages view (pages/businessman/requested_custom_packages).
Test Type	Positive

Test Case ID	BC-05
Description	Edit profile
Precondition	User is logged in as businessman
Test Steps	1. Log in as a businessman user. 2. Send a GET request to /businessman/edit-profile.
Expected Result	Renders the edit profile view (pages/common/edit_profile).
Test Type	Positive

## 6. AuthController

Test Case ID	AUC-01
Description	Login with valid credentials
Precondition	Valid user exists in the database
Test Steps	1. Send a POST request to /login with JSON: {"email": "test@example.com", "password": "valid_password"}.
Expected Result	Returns a JSON response with {"message": "Login successful", "redirect_url": "/dashboard"}, sets session cookie.
Test Type	Positive

Test Case ID	AUC-02
Description	Login with invalid credentials
Precondition	None
Test Steps	1. Send a POST request to /login with JSON: {"email": "invalid@example.com", "password": "wrong"}.
Expected Result	Returns a JSON response with {"error": "Invalid email or password"} and status code 401.
Test Type	Negative

Test Case ID	AUC-03
Description	Login with blocked account
Precondition	Blocked user exists in the database
Test Steps	1. Send a POST request to /login with JSON: {"email": "blocked@example.com", "password": "valid_password"}.
Expected Result	Returns a JSON response with {"error": "Your account is blocked or banned"} and status code 403.
Test Type	Negative

Test Case ID	AUC-04
Description	Logout
Precondition	User is logged in
Test Steps	1. Log in as a user. 2. Send a GET request to /logout.
Expected Result	Redirects to /login, clears session data.
Test Type	Positive

Test Case ID	AUC-05
--------------	--------

Description	Change password with valid current password
Precondition	User is logged in, valid current password
Test Steps	<ol style="list-style-type: none"> <li>1. Log in as a user.</li> <li>2. Send a POST request to /change-password with JSON: {"current_password": "old_password", "new_password": "new_password"}.</li> </ol>
Expected Result	Returns a JSON response with {"message": "Password changed successfully"} and status code 200.
Test Type	Positive

Test Case ID	AUC-06
Description	Change password with invalid current password
Precondition	User is logged in
Test Steps	<ol style="list-style-type: none"> <li>1. Log in as a user.</li> <li>2. Send a POST request to /change-password with JSON: {"current_password": "wrong_password", "new_password": "new_password"}.</li> </ol>
Expected Result	Returns a JSON response with {"error": "Invalid current password"} and status code 401.
Test Type	Negative

Test Case ID	AUC-07
Description	Change password with empty fields
Precondition	User is logged in
Test Steps	<ol style="list-style-type: none"> <li>1. Log in as a user.</li> <li>2. Send a POST request to /change-password with JSON: {}.</li> </ol>
Expected Result	Returns a JSON response with {"error": "Current password and new password are required"} and status code 400.
Test Type	Negative

Test Case ID	AUC-08
Description	Login with non-POST method
Precondition	None
Test Steps	1. Send a GET request to /login.
Expected Result	Returns a JSON response with {"error": "Method Not Allowed"} and status code 405.
Test Type	Negative

## 7. GuestController

Test Case ID	GC-01
Description	Access home page
Precondition	None
Test Steps	1. Send a GET request to /.
Expected Result	Renders the landing page (pages/guest/landing).
Test Type	Positive

Test Case ID	GC-02
Description	Access services list
Precondition	None
Test Steps	1. Send a GET request to /services.
Expected Result	Renders the services list view (pages/guest/services_list).
Test Type	Positive

Test Case ID	GC-03
Description	Access user profile
Precondition	None
Test Steps	1. Send a GET request to /user/1.
Expected Result	Renders the user profile view with user and service data (pages/guest/user_profile).
Test Type	Positive

Test Case ID	GC-04
Description	Access login page when logged in
Precondition	User is logged in as admin
Test Steps	1. Log in as an admin user. 2. Send a GET request to /login.
Expected Result	Redirects to /admin/dashboard.
Test Type	Positive

Test Case ID	GC-05
Description	Access register page when logged in
Precondition	User is logged in as businessman
Test Steps	1. Log in as a businessman user. 2. Send a GET request to /register.
Expected Result	Redirects to /services.
Test Type	Positive

Test Case ID	GC-06
Description	Access 404 error page
Precondition	None
Test Steps	1. Send a GET request to /404.
Expected Result	Renders the 404 error page (pages/status_pages/404).
Test Type	Positive

Test Case ID	GC-07
Description	Access payment success page
Precondition	None
Test Steps	1. Send a GET request to /payment-success.
Expected Result	Renders the payment success page (pages/status_pages/payment_success).
Test Type	Positive

## 8. CommonController Test Cases

Test Case ID	CC-01
Description	Access chat page
Precondition	User is logged in
Test Steps	1. Log in as a user. 2. Send a GET request to /chat.
Expected Result	Renders the chat page (pages/common/chat).
Test Type	Positive



Test Case ID	CC-02
Description	Access chat page without login
Precondition	User is not logged in
Test Steps	1. Send a GET request to /chat.
Expected Result	Redirects to /login.
Test Type	Negative

Test Case ID	CC-03
Description	Access notifications page
Precondition	User is logged in
Test Steps	1. Log in as a user. 2. Send a GET request to /notifications.
Expected Result	Renders the notifications page (pages/common/notifications).
Test Type	Positive

Test Case ID	CC-04
Description	Access notifications without login
Precondition	User is not logged in
Test Steps	1. Send a GET request to /notifications.
Expected Result	Redirects to /login.
Test Type	Negative

## 9. DesignerController

Test Case ID	DC-01
Description	Access dashboard as designer
Precondition	User is logged in as designer
Test Steps	1. Log in as a designer user. 2. Send a GET request to /designer/dashboard.
Expected Result	Renders the seller dashboard view (pages/common/seller_dashboard).
Test Type	Positive

Test Case ID	DC-02
Description	Access dashboard as non-designer
Precondition	User is logged in as non-designer
Test Steps	1. Log in as a non-designer user. 2. Send a GET request to /designer/dashboard.
Expected Result	Redirects to an unauthorized page or returns an error.
Test Type	Negative

Test Case ID	DC-03
Description	Add gig
Precondition	User is logged in as designer
Test Steps	1. Log in as a designer user. 2. Send a GET request to /designer/add-gig.
Expected Result	Renders the add gig view (pages/designer/add_gig).
Test Type	Positive

Test Case ID	DC-04
Description	View earnings
Precondition	User is logged in as designer
Test Steps	1. Log in as a designer user. 2. Send a GET request to /designer/earnings.
Expected Result	Renders the earnings view (pages/common/earnings).
Test Type	Positive

Test Case ID	DC-05
Description	Access payout methods
Precondition	User is logged in as designer
Test Steps	1. Log in as a designer user. 2. Send a GET request to /designer/payout-methods.
Expected Result	Renders the payout methods view (pages/common/payout_methods).
Test Type	Positive

## 10. ActionController

Test Case ID	ACC-01
Description	Get action list as admin
Precondition	User is logged in as admin
Test Steps	1. Log in as an admin user. 2. Send a GET request to /actions?limit=10&page=1.
Expected Result	Returns a JSON response with a paginated list of action records and pagination metadata.
Test Type	Positive

Test Case ID	ACC-02
Description	Get action list as non-admin
Precondition	User is logged in as non-admin
Test Steps	1. Log in as a non-admin user. 2. Send a GET request to /actions.
Expected Result	Returns a JSON response with {"error": "Unauthorized access"} and status code 403.
Test Type	Negative

Test Case ID	ACC-03
Description	Get action list with filters
Precondition	User is logged in as admin, actions exist
Test Steps	1. Log in as an admin user. 2. Send a GET request to /actions?action_type=complaint&date_from=2025-01-01.
Expected Result	Returns a JSON response with filtered action records.
Test Type	Positive

Test Case ID	ACC-04
Description	Get action list with invalid sort_by
Precondition	User is logged in as admin
Test Steps	1. Log in as an admin user. 2. Send a GET request to /actions?sort_by=invalid.
Expected Result	Returns a JSON response with a list sorted by default (created_at).
Test Type	Negative

Test Case ID	ACC-05
--------------	--------

Description	Get action list with large limit
Precondition	User is logged in as admin
Test Steps	1. Log in as an admin user. 2. Send a GET request to /actions?limit=1000.
Expected Result	Returns a JSON response with a large dataset or server handles gracefully without crashing.
Test Type	Edge

## 11. InfluencerController

Test Case ID	INF-01
Description	Verify dashboard access for authenticated influencer
Preconditions	User is logged in with influencer role
Test Steps	1. Call /dashboard endpoint
Expected Result	Returns 200 OK with dashboard HTML
Test Type	Positive

Test Case ID	INF-02
Description	Block dashboard access for non-influencer
Preconditions	User is logged in with businessman role
Test Steps	1. Call /dashboard endpoint
Expected Result	Returns 403 Forbidden
Test Type	Negative

## 12. OrderController

Test Case ID	ORD-01
Description	Create a valid order
Preconditions	Valid service_id, authenticated buyer
Test Steps	1. POST /createOrder with required fields
Expected Result	Returns 201 Created with order_id
Test Type	Positive

Test Case ID	ORD-02
Description	Reject order with invalid payment type
Preconditions	Valid service_id but payment_type=invalid
Test Steps	1. POST /createOrder
Expected Result	Returns 400 Bad Request
Test Type	Negative

### 13. PaymentController

Test Case ID	PAY-01
Description	Successful fund release after hold period
Preconditions	Order completed, hold period expired
Test Steps	1. Call /releaseFunds with valid transaction_id
Expected Result	Returns 200 OK, updates wallet balances

Test Type	Positive
-----------	----------

Test Case ID	PAY-02
Description	Reject fund release before hold period ends
Preconditions	Order completed, hold period not expired
Test Steps	1. Call /releaseFunds prematurely
Expected Result	Returns 400 Bad Request
Test Type	Negative

## 14. OrderDeliveryController

Test Case ID	DEL-01
Description	Seller submits valid delivery
Preconditions	Order status = in_progress, seller logged in
Test Steps	1. POST /createDelivery with files/notes



Expected Result	Returns 200 OK, order status → completed
Test Type	Positive

Test Case ID	DEL-02
Description	Reject empty delivery note
Preconditions	Order status = in_progress
Test Steps	1. POST /createDelivery with empty delivery_note
Expected Result	Returns 400 Bad Request
Test Type	Negative

## 15. RegistrationController

Test Case ID	REG-01
Description	Register with valid email/password
Preconditions	Email not previously registered
Test Steps	1. POST /createUser with valid data
Expected Result	Returns 200 OK, sends verification email
Test Type	Positive

Test Case ID	REG-02
Description	Reject duplicate email registration
Preconditions	Email already exists in DB
Test Steps	1. POST /createUser with duplicate email
Expected Result	Returns 400 Bad Request

Test Type	Negative
-----------	----------

## 16. RevisionController

Test Case ID	REV-01
Description	Buyer requests revision with remaining quota
Preconditions	remained_revisions > 0, delivery exists
Test Steps	1. POST /requestRevision with notes/files
Expected Result	Returns 200 OK, decrements remained_revisions
Test Type	Positive

Test Case ID	REV-02
Description	Reject revision when quota exhausted
Preconditions	remained_revisions = 0
Test Steps	1. POST /requestRevision

Expected Result	Returns 400 Bad Request
Test Type	Negative

## 17. ServiceController.php

Test Case ID	SVC-01
Description	Influencer creates a promotion service
Preconditions	Authenticated influencer, valid files
Test Steps	1. POST <code>/createService</code> with required fields
Expected Result	Returns 201 Created, service visible in listings
Test Type	Positive

Test Case ID	SVC-02
Description	Reject service creation with invalid file type
Preconditions	Authenticated influencer

Test Steps	1. Upload .exe file instead of image
Expected Result	Returns 400 Bad Request
Test Type	Negative

## 18. UserController

Test Case ID	USR-01
Description	Update profile with valid image
Preconditions	Authenticated user
Test Steps	1. POST /updateUserProfile with new profile photo
Expected Result	Returns 200 OK, profile photo updated
Test Type	Positive

Test Case ID	USR-02
--------------	--------

Description	Reject oversized profile image (>5MB)
Preconditions	Authenticated user
Test Steps	1. Upload 6MB image file
Expected Result	Returns 413 Payload Too Large
Test Type	Negative

## 19. VerificationController

Test Case ID	VC-01
Function	<code>getVerificationsList()</code>
Description	Admin successfully fetches verification list with valid query parameters.
Test Steps	Admin role, <code>limit=10</code> , <code>offset=0</code> , <code>sort_by=name</code> , <code>order_dir=asc</code>
Expected Output	JSON response with verification list and pagination data ( <code>200 OK</code> ).
Test Type	Positive

Test Case ID	VC-02
Function	getVerificationsList()
Description	Non-admin attempts to fetch verification list.
Test Steps	Non-admin role (e.g., role=user).
Expected Output	403 Unauthorized error.
Test Type	Negative

Test Case ID	VC-03
Function	getVerificationsList()
Description	Invalid pagination parameters provided.
Test Steps	limit=abc, offset=xyz.
Expected Output	400 Bad Request error.
Test Type	Negative

Test Case ID	VC-04
--------------	-------

Function	updateVerificationStatus()
Description	Admin updates business verification status to "verified".
Test Steps	id=1, type=business, status=verified.
Expected Output	{"success": true, "message": "Verification status updated successfully"} (200 OK).
Test Type	Positive

Test Case ID	VC-05
Function	updateVerificationStatus()
Description	Missing required fields for status update.
Test Steps	id=1, type=business (no status field).
Expected Output	400 Bad Request error.
Test Type	Negative

Test Case ID	VC-06
Function	updateVerificationStatus()



Description	Invalid status value provided.
Test Steps	status=pending (invalid value).
Expected Output	400 Bad Request error.
Test Type	Negative

Test Case ID	VC-07
Function	getVerificationDetails()
Description	Fetch business verification details successfully.
Test Steps	id=1, type=business.
Expected Output	JSON with formatted business details (200 OK).
Test Type	Positive

Test Case ID	VC-08
Function	getVerificationDetails()
Description	Missing required parameters.
Test Steps	No id or type provided.

Expected Output	400 Bad Request error.
Test Type	Negative

## 20. UserController

Test Case ID	UC-01
Function	getUserList()
Description	Fetch user list with valid pagination.
Test Steps	limit=10, offset=0, sort_by=name.
Expected Output	JSON with user list and pagination (200 OK).
Test Type	Positive

Test Case ID	UC-02
Function	getUserList()
Description	Invalid pagination parameters.

Test Steps	limit=abc, offset=xyz.
Expected Output	400 Bad Request error.
Test Type	Negative

Test Case ID	UC-03
Function	getUserProfile()
Description	Fetch businessman profile successfully.
Test Steps	id=1, role=businessman.
Expected Output	JSON with profile and business data (200 OK).
Test Type	Positive

Test Case ID	UC-04
Function	getUserProfile()
Description	Non-existent user ID provided.
Test Steps	id=999.

Expected Output	404 Not Found error.
Test Type	Negative

Test Case ID	UC-05
Function	updateUserProfile()
Description	Update basic user info successfully.
Test Steps	name=John, email=john@test.com.
Expected Output	Success message (200 OK).
Test Type	Positive

Test Case ID	UC-06
Function	updateUserProfile()
Description	Missing required fields for update.
Test Steps	No name or email provided.
Expected Output	400 Bad Request error.
Test Type	Negative

Test Case ID	UC-07
Function	updateUserAccountStatus()
Description	Admin updates user status to "active".
Test Steps	status=active.
Expected Output	Success message (200 OK).
Test Type	Positive

Test Case ID	UC-08
Function	updateUserAccountStatus()
Description	Non-admin attempts to update status.
Test Steps	Non-admin role (e.g., role=user).
Expected Output	401 Unauthorized error.
Test Type	Negative

Test Case ID	UC-09
--------------	-------

Function	updateUserAccountStatus()
Description	Invalid status value provided.
Test Steps	status=invalid.
Expected Output	400 Bad Request error.
Test Type	Negative

Test Case ID	UC-10
Function	deleteUserProfile()
Description	Admin deletes user successfully.
Test Steps	Valid user_id.
Expected Output	Success message (200 OK).
Test Type	Positive

Test Case ID	UC-11
Function	deleteUserProfile()
Description	Non-admin attempts to delete user.

Test Steps	Non-admin role.
Expected Output	401 Unauthorized error.
Test Type	Negative

Test Case ID	UC-12
Function	deleteUserProfile()
Description	Invalid user ID provided.
Test Steps	user_id=999.
Expected Output	404 Not Found error.
Test Type	Negative