

ID	Test Scenario	Method	Endpoint	Request Body (JSON)	Expected Result	Actual results	Status	Comments	Screenshot
TC_API-001	Sign-Up – Success	POST	/signup	{ "username": "TestUser88", "password": "MTlzMzlxTm4h" }	Status 200, empty body ""	200 OK, empty body ""	Pass	User successfully created. Response saved as Example in Postman.	N/A
TC_API-002	Sign-Up – User Already Exists	POST	/signup	{ "username": "TestUser88", "password": "MTlzMzlxTm4h" }	Status 200, { "errorMessage": "This user already exist." }	200 OK, { "errorMessage": "This user already exist." }	Pass	Duplicate user correctly handled. Validation works.	N/A
TC_API-003	Sign-Up – Empty Username	POST	/signup	{ "username": "", "password": "MTlzMzlxTm4h" }	Status 400, { "errorMessage": "Username required" }	500 Internal Server Error	Fail (linked to BUG_API-001)	Server crash on empty username (unhandled input). Linked to BUG_API-001.	Screenshots/API/BUG_API-001_TC_API-005_Sign_up_empty_credentials_error_500.PNG
TC_API-004	Sign-Up – Empty Password	POST	/signup	{ "username": "TestUser89", "password": "" }	Status 400, { "errorMessage": "Password required" }	200 OK, { "errorMessage": "This user already exist." }	Fail (linked to BUG_API-001)	No password validation; backend incorrectly checks duplicate user instead. Linked to BUG_API-001.	Screenshots/API/BUG_API-001_TC_API-005_Sign_up_empty_credentials_error_500.PNG
TC_API-005	Sign-Up – Both Empty	POST	/signup	{ "username": "", "password": "" }	Status 400, { "errorMessage": "Username and password required" }	500 Internal Server Error	Fail (BUG_API-001)	Server crash on empty credentials (unhandled input). Response saved as Example (500 error).	Screenshots/API/BUG_API-001_TC_API-005_Sign_up_empty_credentials_error_500.PNG
TC_API-006	Sign-Up – Invalid Characters	POST	/signup	{ "username": "Test@User", "password": "1234" }	Status 400, { "errorMessage": "Invalid characters" }	Not tested	Fail (known UI issue)	UI test failed and confirmed that API accepts any characters — no character rules applied. Input validation missing globally.	N/A
TC_API-007	Login – Success	POST	/login	{ "username": "TestUser88", "password": "MTlzMzlxTm4h" }	Status 200, success message or token	200 OK, Auth_token: VGVzdFVzZXI4ODE3NjA5NTk=	Pass	Token successfully generated and returned. Authentication endpoint works.	N/A
TC_API-008	Login – Wrong Password	POST	/login	{ "username": "TestUser88", "password": "WrongPass" }	Status 200, { "errorMessage": "Wrong password" }	200 OK, { "errorMessage": "Wrong password." }	Pass	Negative test passed — API returns proper error message.	N/A
TC_API-009	Login – Nonexistent User	POST	/login	{ "username": "NoUser", "password": "MTlzMzlxTm4h" }	Status 200, { "errorMessage": "User does not exist" }	200 OK, { "errorMessage": "User does not exist." }	Pass	Negative test passed — backend correctly handles nonexistent user.	N/A
TC_API-010	Login – Empty Username	POST	/login	{ "username": "", "password": "MTlzMzlxTm4h" }	Status 400, { "errorMessage": "Username required" }	500 Internal Server Error (HTML)	Fail (linked to BUG_API-001)	Backend crash — unhandled empty username. Same issue pattern as Sign-Up (BUG_API-001). Linked to BUG_API-001.	Screenshots/API/BUG_API-001_TC_API-005_Sign_up_empty_credentials_error_500.PNG
TC_API-011	Login – Empty Password	POST	/login	{ "username": "TestUser88", "password": "" }	Status 400, { "errorMessage": "Password required" }	200 OK, { "errorMessage": "Wrong password." }	Pass	Negative login scenario handled correctly. API returns appropriate error message.	N/A
TC_API-012	Logout – Success	POST	/logout	Empty request body	Status 200, { "message": "Logout successful" }	404 Not Found	N/A	/logout endpoint not implemented on Demoblaze API. Logout likely handled client-side (UI only).	N/A
TC_API-013	Response Format Validation	Any	Any	N/A	Valid JSON, contains either success or errorMessage	/signup: valid JSON → OK; /login: plain text token (no JSON keys); /logout: non-JSON HTML	Partial Pass	Known issue: inconsistent response formats across endpoints (violates API response structure standard). No bug raised for demo API. Would be a bug in a real project.	N/A
TC_API-014	Response Time	Any	Any	N/A	< 2 seconds	All requests completed under 2 s (verified in Postman)	Pass	Performance acceptable for all endpoints	N/A
TC_API-015	Environment Variables Test	Any	Any	N/A	Variables {{base_url}}, {{username}}, {{password}} resolve correctly	All requests used correct base URL (https://api.demoblaze.com); variable {{base_url}} resolved successfully from environment and collection.	Pass	Environment and collection variables configured and active as expected	N/A
TC_API-016	Test Script Validation	Any	Any	N/A	Postman test confirms correct success/error message	Collection-level script executed for all requests; detected 500/404 errors and skipped non-JSON cases correctly	Pass	Automation script works as intended; accurately reports server errors and response type	N/A
TC_API-017	Add item to cart – success	POST	/addtocart	{ "id": "fa1c9918-875d-c378-3035-f5c2a772f861", "cookie": "VGZzdFVzZXI4MDE3NjEyMDE=", "prod_id": 2, "flag": true }	200 OK	200 OK	Pass	Validated with real token and product ID captured from DevTools. Response 200 OK, empty body (expected for demo API).	N/A
TC_API-018	Add to Cart – Invalid Product ID	POST	/addtocart	{ "id": "fa1c9918-875d-c378-3035-f5c2a772f861", "cookie": "VGZzdFVzZXI4MDE3NjEyMDE=", "prod_id": 999999, "flag": true }	Status 400 or error message { "errorMessage": "Invalid product ID" }	Status 200 OK, empty body; response time 7207 ms	Fail (BUG_API-002)	Product ID not validated by backend; server accepted invalid product. Response time exceeded 2 s limit.	Screenshots/API/BUG_API-002_TC_API-018_Server_accepted_invalid_product_id.PNG
TC_API-019	View Cart – Success	POST	/viewcart	{ "cookie": "VGZzdFVzZXI4MDE3NjEyMDE=", "flag": true }	200 OK, JSON list of items	Status 200 OK, { "items": [] }	Partial Pass (Environment limitation – expected behavior in demo setup)	Response successful but cart data not persisted outside browser session (demo API limitation). Cart data is session-based — items are visible only during the active UI session and disappear after page refresh or navigation. Verified that Demoblaze demo backend maintains cart state only within an active browser session; the token is not reusable externally. Expected behavior for the demo environment — no bug raised.	N/A
TC_API-020	Delete item from cart – success	POST	/deleteitem	{ "id": "8fbeb8ed-ce60-d0d0-1bffe856139fd3" }	Status 200 OK, JSON { "success": true }	Status 200 OK, { "errorMessage": "Not found." }	Partial Pass (Environment limitation – expected behavior in demo setup)	Response successful but cart data not persisted outside browser session (demo API limitation). Cart data is session-based — items are visible only during the active UI session and disappear after page refresh or navigation. Verified that Demoblaze demo backend maintains cart state only within an active browser session; the token is not reusable externally. Expected behavior for the demo environment — no bug raised.	N/A
TC_API-021	Place order – success	POST	/placeorder	{ "name": "Test User", "country": "Israel", "city": "Tel Aviv", "creditCard": "1234-5678-9999-0000", "month": "10", "year": "2025" }	200 OK, order ID returned (e.g., { "orderid": 12345 })	404 Not Found	Partial Pass (Environment limitation – expected behavior in demo setup)	Endpoint not implemented in demo backend; checkout simulated only on UI. Expected limitation of demo API — no bug raised.	N/A
TC_API-022	Place order – missing payment details	POST	/placeorder	{ "name": "Test User" }	400, { "errorMessage": "Payment details required" }	404 Not Found	Partial Pass (Environment limitation – expected behavior in demo setup)	Endpoint not implemented in demo backend; checkout simulated only on UI. Expected limitation of demo API — no bug raised.	N/A
TC_API-023	Place order – invalid card format	POST	/placeorder	{ "creditCard": "abcd" }	400, { "errorMessage": "Invalid card number" }	Not tested	Not Tested	Endpoint not implemented; all negative cases return 404 Not Found — same as TC_API-022.	N/A

TC_API-024	Response validation	Any	Any	N/A	Valid JSON with success or errorMessage; under 2s response	All requests passed validation; 1 skipped due to HTML (non-JSON) response (known backend issue), all others valid and < 2 s	Pass	Automation script works as intended; accurately reports server errors and response type	N/A
------------	---------------------	-----	-----	-----	--	---	------	---	-----