Bachao.AI

BACHAO.AI · SECURITY ASSESSMENT REPORT

G****** — Web, API & Mobile Security Assessment (Sample · Client Identity Redacted)

Management platform (public-sector healthcare-logistics tenant) — API, dashboard & Android apps (consolidated interim)

Sample deliverable — a real Bachao.AI engagement with the client name, product name, domains and package names redacted (asterisks / generic placeholders); the findings, evidence and structure are the genuine report. Authorized VAPT of the backend API and web dashboard (black-box + authenticated grey-box) and static/code-level MASVS/MASTG review of two Android apps. Consolidated interim report — mobile dynamic testing and web write-endpoint retest remain.

TLP:AMBER · Confidential

Prepared for: **G******

Prepared by: **Bachao.AI — Dhisattva AI Pvt Ltd (DPIIT-Recognised Startup)**

Assessment: **August 2026**

Certificate No.: **BVPT-SAMPLE-2026-0003** · Classification: **TLP:AMBER — Confidential**

◆ Findings require remediation

14 findings identified — 2 Critical · 3 High · 5 Medium · 4 Low. Prioritised remediation guidance is provided.

Sample deliverable — do not scan

This is a masked demonstration report; the certificate reference is **not a live verification**. Real client certificates carry a scannable, OTP-authenticated QR at **bachao.ai/verify**.

Contents

The report is organised as follows.

1 Document Control

Report title	G****** — Web, API & Mobile Security Assessment (Sample · Client Identity Redacted)
Client	G****** Technologies
Certificate / Reference No.	BVPT-SAMPLE-2026-0003

Engagement	Management platform (public-sector healthcare-logistics tenant) — API, dashboard & Android apps (consolidated interim)
Assessment type	Web/API: Black-box + Authenticated Grey-box (OWASP API Top 10, PTES, NIST SP 800-115) · Mobile: OWASP MASVS/MASTG static & code review
Assessment window	August 2026
Report version	1.0 — Final
Classification	CONFIDENTIAL — restricted to G****** & Bachao.AI
Assessor	Bachao.AI — Dhisattva AI Pvt Ltd (DPIIT-Recognised Startup)
Standards	CERT-In-aligned · ISO/IEC 27001:2022 · SOC 2 · GDPR · DPDP · OWASP · CVSS v3.1
Lead assessor	Shouvik Mukherjee — Lead Security Assessor
Verification	sample deliverable — reference not live

2 About This Assessment

Bachao.AI (Dhisattva AI Pvt Ltd) is a DPIIT-recognised cybersecurity and fraud-prevention company. This engagement was an authorised Vulnerability Assessment & Penetration Test (VAPT) of G******'s internet-facing production surface, with prioritised remediation guidance. Testing combined unauthenticated black-box probing with authenticated grey-box testing (capture-and-replay of the application's real API calls under a valid session). The assessment was conducted to a **CERT-In-aligned methodology** and its findings are mapped to **ISO/IEC 27001:2022, SOC 2, GDPR and the DPDP Act 2023** control frameworks, following the **OWASP** Top 10 and API Security Top 10 testing standards and scored with **CVSS v3.1** — meeting Indian and international security-assessment benchmarks.

This report is confidential and intended for G******. It may be shared with the client's customers, investors, auditors and regulators as independent evidence of the security testing performed.

3 Executive Summary

This is a sample deliverable: a real Bachao.AI engagement with the client name, product name, domains and Android package names redacted to protect confidentiality — the findings, evidence and structure are the genuine report. The assessment found that authentication on the platform is fundamentally sound — every API endpoint rejects unauthenticated and forged (alg:none) requests, and no SQL injection or credential leak was confirmed — but authorization is broken across the board. Any single authenticated user, at the lowest privilege level, can read every record's data and personal information by enumerating sequential object IDs (F-01), invoke every administrative function including password reset and user creation (F-02), and tamper with inventory records they do not own (F-03). Together these constitute a critical, systemic authorization failure that exposes bulk personal data (PAN, phone, GPS) with direct DPDP Act implications. Note: the automated black-box pass reported two 'critical' SQL-injection/secret findings that were false positives caused by the single-page app returning HTTP 200 for all paths; these were verified and reclassified (see the false-positive register) and are not counted as findings. On the mobile side, a static/code-level MASVS review of the two Android apps found the WMS app shipped as a debug build (debuggable + debug-signed, with no runtime hardening), no TLS certificate pinning, and — most importantly — the app consuming the same object-ID API endpoints, so mobile users inherit the same BOLA/BFLA exposure (MOB-01/02/03). Mobile dynamic testing (runtime, TLS interception, exported-component) and the web write-endpoint retest remain; the mobile builds reviewed were debug builds, so a production release should be re-checked. No certificate of secure posture is issued until the authorization defects are remediated and retested.

2	3	5	4	0
CRITICAL	HIGH	MEDIUM	LOW	INFO

Posture: prioritise the Critical/High findings first; the roadmap in this report sequences remediation by risk.

4 Scope of Work

The following assets were in scope. Testing was non-destructive and read-only: no data was deleted or altered, no active denial-of-service or load testing was performed, and no real personal data was exfiltrated. All activity is recorded in a timestamped forensic log.

HOST	ROLE	RESOLVES TO	STATUS
`api.b*******.com`	Backend API (/v1/api)	—	In scope
`dashboard.b*******.com`	Web Dashboard (SPA)	—	In scope
`com.example.wms`	Android app — WMS (static/code)	—	In scope
`com.example.companion`	Android app — companion/video (static/code)	—	In scope

Authenticated testing. Authenticated grey-box performed as a low-privilege operator (role_id=2) using a client-provisioned test account. Testing was non-destructive and read-only against production; mutating endpoints were exercised only via safe authorization probes (empty/invalid bodies, non-existent IDs). One orphaned test record created inadvertently during testing was flagged to the client for removal.

Consolidated interim across all 4 paid assets: Backend API + Web Dashboard (black-box + authenticated grey-box) and 2 Android apps (MASVS/MASTG static & code review). OPEN: (a) mobile DYNAMIC testing (runtime/MITM/exported-component) pending a device + a production release APK — the builds reviewed were debug builds; (b) web write-endpoint authz was verified via safe probes, destructive-only checks deferred to staging; (c) CVSS vectors pending final peer review. One orphaned test record created during authorized testing was flagged to the client for removal.

5 Methodology

The engagement follows **OWASP ASVS Levels 1–2**, the **PTES** phases and **NIST SP 800-115** technical testing guidance, aligned to **CERT-In** reporting expectations.

1 · Reconnaissance & asset discovery

Passive discovery (Certificate Transparency, DNS) and active enumeration of the in-scope hosts, sub-domains, API base URLs and technology stack; TLS/transport and email-security posture.

2 · Automated assessment (curated, AI-native)

Our platform runs a curated suite covering the OWASP Top 10 and OWASP API Security Top 10 in isolated micro-VMs; every candidate finding is re-validated by a second pass to strip false positives before a human sees it.

3 · Authenticated grey-box (capture-replay)

Real sessions at each privilege level are captured and replayed with access-control tampering (BOLA/IDOR, function-level authorization, mass-assignment) — the class of flaw black-box structurally cannot find.

4 · Senior-led manual exploitation

Hands-on verification of business logic, tenant isolation and result-integrity abuse; every reported finding is manually reproduced with request/response evidence.

5 · Risk rating & reporting

Each finding is scored with CVSS v3.1, mapped to OWASP/CWE and the client's compliance frameworks, and written with a concrete, developer-ready remediation.

Tooling by phase

PHASE	REPRESENTATIVE TOOLING / CHECKS
Reconnaissance	Certificate Transparency (crt.sh), DNS / DNS-over-HTTPS, port & service enumeration, TLS inspection
Transport & TLS	openssl s_client, cipher/protocol & certificate-chain analysis
Email security	SPF / DKIM / DMARC record analysis via authoritative DoH lookups
Web & API	OWASP Top 10 / API Top 10 curated suite, HTTP method & header inspection, CORS probing
Authenticated	Session capture-replay harness with BOLA/IDOR/BFLA and mass-assignment tampering
Verification	Manual reproduction with curl / HTTP proxy; CVSS v3.1 scoring

6 Risk Rating Methodology

Severity is derived from the **CVSS v3.1** base score, adjusted by the assessor for business context (data sensitivity, exposure, exploitability).

SEVERITY	CVSS BAND	MEANING & EXPECTED ACTION
CRITICAL	9.0 – 10.0	Immediate exploitation risk; direct compromise of data or systems. Fix immediately.
HIGH	7.0 – 8.9	Serious impact; realistic exploitation path. Fix urgently.
MEDIUM	4.0 – 6.9	Moderate impact or conditional exploitation. Fix in the normal cycle.
LOW	0.1 – 3.9	Limited impact; defense-in-depth / hardening.
INFO	0.0	No direct security impact; informational / good-practice note.

7 Test Coverage

The assessment executed **620 individual checks** across the categories below; “observations” are candidate signals raised for analyst review (the confirmed subset, after verification, appears in §Detailed Findings).

CATEGORY	WHAT IT CHECKS	CHECKS	OBS.
API Security (OWASP API Top 10)	Object- and function-level authorization, property-level exposure, auth, rate limiting.	200	6
Authentication & Session	Token design/lifetime, signature enforcement, session storage, forgery resistance.	90	3

CATEGORY	WHAT IT CHECKS	CHECKS	OBS.
Authorization (BOLA/BFLA)	Cross-object and cross-function access as a low-privilege operator; ownership checks.	60	3
Mobile (OWASP MASVS/MASTG — static/code)	Build hardening, TLS pinning, inherited API authz, permissions, exported components, storage.	80	6
Reconnaissance & Discovery	Subdomain/asset discovery, DNS/TLS posture, technology fingerprinting, exposed services.	120	6
Security Misconfiguration	Verbose errors, embedded keys, DNS hygiene, response headers.	70	4
Total		**620**	**28**

8 Findings Summary

ID	SEVERITY	FINDING	HOST	CVSS
F-01	CRITICAL	Broken Object-Level Authorization (BOLA/IDOR) — any authenticated user can read every record's data and PII by sequential ID	api.b*******.com	9.3
F-02	CRITICAL	Broken Function-Level Authorization (BFLA) — a low-privilege operator can reach all administrative functions	api.b*******.com	9.1
F-03	HIGH	Broken write authorization — update endpoints modify arbitrary objects with no ownership or existence check	api.b*******.com	8.1
MOB-01	HIGH	Debug build shipped — debuggable + debug-signed, with no runtime protection	Android — com.example.wms	7.1
MOB-03	HIGH	Mobile clients inherit the API authorization defects (BOLA/BFLA)	Android — com.example.wms (→ api.b*******.com)	7.1
F-04	MEDIUM	Excessive data exposure & user enumeration — operator can list all users including administrators	api.b*******.com	6.5
F-05	MEDIUM	Weak session-token design — long-lived HS256 JWT with no role/tenant claim, stored in localStorage	dashboard.b*******.com / api.b*******.com	5.3
F-06	MEDIUM	Unrestricted Google Maps API key embedded in the client bundle	dashboard.b*******.com	5.3
MOB-02	MEDIUM	No TLS certificate pinning / no network security config	Android — com.example.wms / com.example.companion	5.9
MOB-04	MEDIUM	Over-broad Android permissions (all-files access, camera, microphone)	Android — com.example.wms / com.example.companion	5.0
F-07	LOW	Verbose backend/database error messages returned to clients	api.b*******.com	4.3
F-08	LOW	Low DNS TTL may facilitate DNS rebinding	api.b*******.com	3.7

ID	SEVERITY	FINDING	HOST	CVSS
MOB-05	LOW	Unintended exported activity	Android — com.example.companion	3.3
MOB-06	LOW	Session token handled via Intent extras / in-memory (verify at runtime)	Android — com.example.wms	3.3

9 Detailed Findings

F-01 CRITICAL ● OPEN

Broken Object-Level Authorization (BOLA/IDOR) — any authenticated user can read every record's data and PII by sequential ID

api.b*******.com · /v1/api/dashboard/get-mapp-data · get-pending-items · get-expiry-count · /v1/api/rep/get

CVSS 9.3 · AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N · API1:2023 — Broken Object Level Authorization · CWE-639

Background. Object-level authorization is the most common and most damaging API flaw: the server must check, on every request, that the authenticated caller is actually permitted to see the specific object being requested. When the object is addressed by a guessable identifier and no ownership check is performed, any user can enumerate the identifier space and read every object in the system.

Description. The object endpoints accept a numeric object identifier (`objectId` / `object_id`) and return the full record without verifying that the caller owns or is scoped to that object. Authenticated as a single low-privilege operator (role_id=2), iterating `objectId=1..N` returned HTTP 200 with the complete data of every existing object (~287–313 KB each), while non-existent IDs returned an empty 73-byte body — a clean enumeration oracle. The behaviour is identical across get-mapp-data, get-pending-items, get-expiry-count and rep/get.

Impact. Any single authenticated user can exfiltrate the entire dataset for all deployments — inventory (SKU/kit/batch/expiry) and personal data including PAN, phone, name and GPS location — by walking sequential IDs. For a healthcare-logistics platform this is a bulk personal-data breach with direct DPDP Act exposure. CVSS base is raised to scope-changed because the authorization flaw in one user context exposes data belonging to every other tenant/deployment.

Evidence.

```
# authenticated as VAPT_TEST (role_id=2); values redacted, structure/status only
GET /v1/api/dashboard/get-mapp-data?objectId=1    -> 200  bodylen=287035
GET /v1/api/dashboard/get-mapp-data?objectId=2    -> 200  bodylen=290660
GET /v1/api/dashboard/get-mapp-data?objectId=50   -> 200  bodylen=289834
GET /v1/api/dashboard/get-mapp-data?objectId=9999 -> 200 bodylen=73 (empty = no such object)
GET /v1/api/dashboard/get-pending-items?objectId=2 -> 200
GET /v1/api/rep/get?object_id=2                   -> 200
exposed field paths incl: batch_no_sr_no, sku_code, kitname, exp, plus PII (PAN, phone, name, GPS
lat)
```

Remediation. Enforce object-level authorization on every object endpoint: derive the permitted object set from the authenticated principal (tenant/role/assignment) server-side and reject requests for objects outside it with HTTP 403 — never trust a client-supplied ID alone. Prefer unguessable identifiers (UUIDv4) as defense-in-depth. Add automated tests that a second, unrelated account receives 403 for the first account's object IDs.

Compliance. ISO 27001 A.8.3 (information access restriction) · SOC 2 CC6.1 / CC6.3 · DPDP S.8(4) reasonable security safeguards; S.8(3) accuracy — bulk PII exposure

References. OWASP API Security Top 10 — API1:2023 · MITRE CWE-639 · OWASP ASVS 4.0 §4.2

F-02 CRITICAL OPEN

Broken Function-Level Authorization (BFLA) — a low-privilege operator can reach all administrative functions

api.b*******.com · /v1/api/user/create-user · update-user · update-password · active-deactive · /v1/api/rep/approved · rep/complete

CVSS 9.1 · AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:N · API5:2023 — Broken Function Level Authorization · CWE-285

Background. Function-level authorization ensures that privileged operations (user administration, approvals) are reachable only by roles entitled to perform them. When the server authenticates a request but does not check the caller's role before executing an administrative handler, any logged-in user can invoke administrator functionality.

Description. Probed with a low-privilege operator token (role_id=2) using empty/invalid bodies and non-existent identifiers, every administrative endpoint returned a validation (400) or not-found (404) response — never an authorization denial (403). This proves the caller passed the function-level authorization gate on each administrative handler; only input validation or record existence stopped the request. The most serious is `user/update-password`, which an operator can reach to reset other users' passwords, and `user/create-user`, which would allow creating accounts (including role_id=1 administrators).

Impact. A single low-privilege account can escalate to full administrative control: create administrator users, reset any user's password (account takeover, including admins), deactivate arbitrary users (denial of service), and self-approve/complete replenishment workflows. Combined with F-01 this yields full read + administrative compromise of the platform from any authenticated session.

Evidence.

```
# role_id=2 token, empty/invalid bodies & non-existent IDs — NONE returned 403 (authz denial)
POST /v1/api/user/create-user      {}                 -> 400 "Name is required"
POST /v1/api/user/update-password  {user_id:<bogus>}  -> 400 "LoginID is required"  (reached;
would reset arbitrary pw)
POST /v1/api/user/active-deactive  {user_id:<bogus>}  -> 404 "No user found to update"
POST /v1/api/rep/approved          {id:999999999}     -> 404 "Replenishment request not found"
=> every admin handler passes the authz gate for a low-privilege operator (no 403 anywhere)
```

Remediation. Enforce deny-by-default function-level authorization server-side: gate every administrative handler on the caller's role BEFORE any processing, returning 403 for unauthorized roles. Do not rely on the client-side role or on the UI hiding functions. Centralise authorization in middleware and add regression tests asserting an operator token receives 403 on every admin route.

Compliance. ISO 27001 A.8.2 (privileged access rights) · SOC 2 CC6.1 / CC6.3 · DPDP S.8(4) reasonable security safeguards

References. OWASP API Security Top 10 — API5:2023 · MITRE CWE-285 · MITRE CWE-862 (Missing Authorization)

F-03 HIGH ● OPEN

Broken write authorization — update endpoints modify arbitrary objects with no ownership or existence check

api.b*******.com · /v1/api/object/update-object · update-qty-data

CVSS 8.1 · AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:H/A:L · API1:2023 / API5:2023 — Broken Object/Function Level Authorization · CWE-639

Background. Write operations require the same object-level authorization as reads. A handler that reports success without confirming the target exists and belongs to the caller allows integrity attacks against arbitrary records.

Description. `object/update-object` and `object/update-qty-data`, invoked by a low-privilege operator against a non-existent object ID, returned HTTP 200 with "...updated Successfully". The endpoints neither enforce ownership nor verify existence before reporting success, indicating a low-priv user can alter inventory quantities and records for objects they do not own (integrity/inventory tampering). Confirmed non-destructively against a non-existent ID so no real record was changed.

Impact. Authenticated users can tamper with inventory quantities and records across the platform, corrupting medical-supply stock data — a data-integrity and operational-safety risk. The misleading success response also masks the tampering.

Evidence.

```
POST /v1/api/object/update-object    {object_id:999999999} -> 200 "Object updated Successfully"
POST /v1/api/object/update-qty-data  {id:999999999}        -> 200 "Item Quantity Updated
Successfully"
(targets a non-existent ID → proves no ownership/existence check; no real row was modified)
```

> **Remediation.** Verify object existence and caller ownership before any write; return 404/403 appropriately and only report success when a row the caller owns was actually updated (assert affected-row count).

Compliance. ISO 27001 A.8.3 (information access restriction) · SOC 2 CC6.1 · DPDP S.8(3)/S.8(4) accuracy & reasonable safeguards

References. OWASP API1/API5:2023 · MITRE CWE-639 · MITRE CWE-284

MOB-01 HIGH ● OPEN

Debug build shipped — debuggable + debug-signed, with no runtime protection

Android — com.example.wms · AndroidManifest / signing block

CVSS 7.1 · AV:L/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N · M8: Security Misconfiguration (OWASP MASVS-RESILIENCE) · CWE-489

Background. A release Android app must be non-debuggable and signed with the production key. A debuggable, debug-signed build lets anyone attach a debugger, read and modify process memory, and extract in-memory secrets/tokens at runtime — and the app carries no root/debugger/tamper detection of its own to resist it.

Description. `com.example.wms` sets `android:debuggable=true` and is signed with the generic 'Android Debug' certificate (CN=Android Debug, O=Android). Source review confirmed the app implements no anti-debug / anti-root / anti-Frida controls.

Impact. An attacker with the app on any device (or a malicious app) can attach a debugger, dump the session JWT and business data from memory, and manipulate app logic — undermining all client-side trust.

Evidence.

```
AndroidManifest: android:debuggable=true
signing cert: CN=Android Debug, O=Android, C=US (v2 scheme)
no isDebuggerConnected / RootBeer / Frida-detection in app sources
```

> **Remediation.** Ship a release build: `android:debuggable=false`, signed with the production upload/app-signing key; enforce in CI (fail the pipeline on a debuggable/debug-signed artifact).

Compliance. ISO 27001 A.8.25 (secure development) · SOC 2 CC8.1 · DPDP S.8(4) reasonable safeguards

References. OWASP MASTG-TEST-0089 (debuggable) · MITRE CWE-489 · OWASP MASVS-RESILIENCE-2

MOB-03 HIGH ● OPEN

Mobile clients inherit the API authorization defects (BOLA/BFLA)

Android — com.example.wms (→ api.b*******.com) · ApiService: getAllData(objectId), GetItemDetailByPackEpc(epc), postGetObjectId, postShortKits

CVSS 7.1 · AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:L/A:N · API1/API5:2023 (via mobile client) · CWE-639

Background. A mobile app is only as safe as the API behind it. Where the server does not enforce object/function-level authorization, every mobile user inherits that exposure.

Description. The app's `ApiService` calls the exact object-ID endpoints proven vulnerable server-side (`getAllData(@Header Authorization, @Query objectId)`, `GetItemDetailByPackEpc(@Query epc)`, etc.) with `"Bearer "+token`. A mobile user can therefore read/act on records belonging to others (the same BOLA/BFLA as F-01/F-02).

Impact. Any authenticated mobile user can enumerate and read all deployments' data (incl. PAN/phone/GPS) and reach administrative functions — identical to the web exposure.

Evidence.

```
ApiService @Query("objectId") / @Query("epc") reached with Bearer token; BASE_URL
api.b*******.com/v1/api/.
```

> **Remediation.** Fix is server-side — enforce object/function-level authorization on the API (see F-01, F-02). No client change alone can remediate this.

Compliance. ISO 27001 A.8.3 (access restriction) · SOC 2 CC6.1 · DPDP S.8(4) reasonable safeguards

References. OWASP API Security Top 10 — API1/API5:2023

F-04 MEDIUM ● OPEN

Excessive data exposure & user enumeration — operator can list all users including administrators

api.b*******.com · /v1/api/user/get-users

CVSS 6.5 · AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N · API3:2023 — Broken Object Property Level Authorization · CWE-200

Background. Directory endpoints should return only the properties and records the caller is entitled to. Exposing the full user list — including which accounts are administrators — hands an attacker a ready-made target list for credential attacks.

Description. `user/get-users`, called by a low-privilege operator, returns all 76 users with login_id, name, role_id, active status and creation date — including the two administrator accounts (role_id=1). (The response includes a `password` property, but it is empty for all users — no credential leak.)

Impact. An attacker with any account obtains the complete administrator username list and organisation directory, enabling targeted password-reset abuse (see F-02) and phishing.

Evidence.

```
GET /v1/api/user/get-users (role_id=2) -> 200; data[] length = 76;
role_id distribution {3:73, 2:1, 1:2 (admins)}; password field EMPTY for all (no hash leak)
```

> **Remediation.** Restrict user-directory listing to administrative roles; for non-admins return only the properties/ records they require. Never include a password property in any response.

Compliance. ISO 27001 A.8.3 · SOC 2 CC6.1 · DPDP S.8(4) reasonable security safeguards

References. OWASP API3:2023 · MITRE CWE-200

F-05 MEDIUM ● OPEN

Weak session-token design — long-lived HS256 JWT with no role/tenant claim, stored in localStorage

dashboard.b*******.com / api.b*******.com · Authorization: Bearer <JWT>

CVSS 5.3 · AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:N · API2:2023 — Broken Authentication · CWE-613

Background. Bearer tokens should be short-lived, bound to authorization context server-side, and stored where script injection cannot read them. Long-lived tokens widen the window for stolen-token abuse.

Description. The session JWT uses HS256 with a payload of only {exp, user_id} — no role or tenant claim, so authorization decisions cannot be made from the token and instead rely on client-supplied role data (compounding F-01/02). The token has a ~90-day lifetime with no observed rotation and is stored in browser localStorage, making it exfiltratable by any XSS. (Positively, `alg:none` forgery and unsigned tokens are correctly rejected.)

Impact. A stolen token is valid for ~3 months; localStorage storage exposes it to XSS theft; the missing role claim means the server has no trustworthy basis for authorization, reinforcing the BOLA/BFLA flaws.

Evidence.

```
JWT header alg=HS256; payload={"exp":<+~90d>,"user_id":"VAPT_TEST"} — no role/tenant claim
stored in localStorage['user'].token; alg:none forgery -> 401 (correctly rejected)
```

Remediation. Shorten token lifetime (e.g. 15–60 min) with refresh-token rotation; include a server-signed role/tenant claim and enforce authorization from it server-side; store tokens in httpOnly, Secure, SameSite cookies rather than localStorage; consider RS256 with key rotation.

Compliance. ISO 27001 A.8.24 (use of cryptography) · SOC 2 CC6.1 · DPDP S.8(4) reasonable safeguards

References. OWASP API2:2023 · MITRE CWE-613 (Insufficient Session Expiration) · MITRE CWE-522

F-06 MEDIUM OPEN

Unrestricted Google Maps API key embedded in the client bundle

dashboard.b*******.com · JavaScript bundle (client-side)

CVSS 5.3 · AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N · A05:2021 — Security Misconfiguration · CWE-1188

Background. Google Maps browser keys are necessarily public, so the control is not secrecy but restriction — an unrestricted key can be lifted from the bundle and used elsewhere, billing the owner.

Description. A live Google Maps API key is embedded in the dashboard's JavaScript bundle and is not restricted by HTTP referrer/API — it returned a successful Maps response from an unrelated origin. (This is the true, re-rated form of the automated scanner's inflated 'hardcoded secret CVSS 9.1' finding; a publishable Maps key is not a backend secret.)

Impact. Third parties can use the key from any origin, consuming the client's Google Maps quota and billing (financial abuse); no direct application compromise.

Evidence.

```
Google Maps browser key ***REDACTED*** present in bundle; Maps API returns 200 from a foreign origin (no referrer restriction)
```

> **Remediation.** Add HTTP-referrer and API restrictions to the key in Google Cloud console, scope it to the Maps APIs actually used, set a quota cap and billing alerts, and rotate the exposed key.

Compliance. ISO 27001 A.8.9 (configuration mgmt) · SOC 2 CC6.1 · DPDP —

References. Google Maps Platform — API key restrictions · MITRE CWE-1188

MOB-02 MEDIUM OPEN

No TLS certificate pinning / no network security config

Android — com.example.wms / com.example.companion · Retrofit/OkHttp client

CVSS 5.9 · AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:N/A:N · M5: Insecure Communication (OWASP MASVS-NETWORK) · CWE-295

Background. Bearer tokens carried over TLS should be protected by certificate pinning so a rogue or mis-issued CA cannot silently intercept traffic and steal the session.

Description. Source review confirmed the app builds Retrofit with a default OkHttp client (`new Retrofit.Builder().baseUrl("https://api.b*******.com/v1/api/")...build()`) — no `CertificatePinner`, no custom `TrustManager`, and no `networkSecurityConfig` in the manifest. The bundled okhttp `CertificatePinner` class is unused.

Impact. On a network the attacker controls (rogue Wi-Fi, mis-issued/compromised CA, MDM CA), the long-lived bearer JWT and all API traffic can be intercepted and the session hijacked.

Evidence.

```
ApiHelper: default Retrofit/OkHttp, no pinning; AndroidManifest: no networkSecurityConfig.
```

> **Remediation.** Add a `network_security_config.xml` pinning the `api.b*******.com` leaf/intermediate (with backup pins + expiry), and/or configure OkHttp `CertificatePinner`. Never weaken hostname verification.

Compliance. ISO 27001 A.8.24 (cryptography) · SOC 2 CC6.7 · DPDP S.8(4) reasonable safeguards

References. OWASP MASVS-NETWORK-1 · MITRE CWE-295

MOB-04 MEDIUM ● OPEN

Over-broad Android permissions (all-files access, camera, microphone)

Android — com.example.wms / com.example.companion · AndroidManifest permissions

CVSS 5.0 · AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N · M6: Inadequate Privacy Controls (OWASP MASVS-PRIVACY) · CWE-250

Background. Least privilege limits the blast radius of a compromised app and satisfies data-minimisation duties. `MANAGE_EXTERNAL_STORAGE` (All-Files-Access) is Play-restricted and grants read/write over the whole shared storage.

Description. Both apps request `MANAGE_EXTERNAL_STORAGE` + READ/WRITE_EXTERNAL_STORAGE; `com.example.wms` also requests CAMERA, RECORD_AUDIO and FINE/COARSE_LOCATION.

Impact. Excess access to device storage and sensors increases privacy risk and the impact of any compromise; RECORD_AUDIO in particular needs justification for a WMS/inventory app.

Evidence.

```
Manifest: MANAGE_EXTERNAL_STORAGE, READ/WRITE_EXTERNAL_STORAGE, CAMERA, RECORD_AUDIO, ACCESS_FINE/
COARSE_LOCATION.
```

> **Remediation.** Drop permissions not tied to a feature (justify CAMERA=barcode scan; review RECORD_AUDIO/ location); migrate off MANAGE_EXTERNAL_STORAGE to scoped storage / MediaStore / SAF.

Compliance. ISO 27001 A.8.3 · SOC 2 CC6.1 · DPDP S.8(1) purpose limitation / data minimisation

References. OWASP MASVS-PRIVACY-1 · MITRE CWE-250 · Google Play — All files access policy

F-07 LOW ● OPEN

Verbose backend/database error messages returned to clients

api.b*******.com · /v1/api/object/* (error responses)

CVSS 4.3 · AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N · A05:2021 — Security Misconfiguration · CWE-209

Background. Detailed internal errors help an attacker fingerprint the stack and craft further attacks.

Description. Several API responses leak raw database error text such as `sql: no rows in result set`, revealing the persistence layer and query behaviour.

Impact. Information disclosure aiding backend fingerprinting and further attack crafting.

Evidence.

```
POST /v1/api/object/create-data {} -> 400 "Error Creating record >sql: no rows in result set"
POST /v1/api/object/add-object-data {} -> 500 "Failed to fetch existing data: sql: no rows in
result set"
```

> **Remediation.** Return generic client-facing error messages; log detailed errors server-side only.

Compliance. ISO 27001 A.8.9 · SOC 2 CC6.1 · DPDP —

References. MITRE CWE-209

F-08 LOW OPEN

Low DNS TTL may facilitate DNS rebinding

api.b*******.com · DNS

CVSS 3.7 · AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N · A05:2021 — Security Misconfiguration · CWE-350

Background. Very low DNS TTLs can be abused in rebinding attacks against clients that resolve and trust the name.

Description. The API host publishes a DNS TTL of 1 second.

Impact. Marginal; defense-in-depth. Rebinding requires additional client-side preconditions.

Evidence.

```
DNS_TTL:1 on api.b*******.com
```

> **Remediation.** Raise the TTL to a normal operational value unless a low TTL is required for failover.

Compliance. ISO 27001 A.8.9 · SOC 2 CC6.1 · DPDP —

References. MITRE CWE-350

MOB-05 LOW OPEN

Unintended exported activity

Android — com.example.companion · AndroidManifest — exported activity

CVSS 3.3 · AV:L/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N · M4: Insufficient Input/Output Validation (OWASP MASVS-PLATFORM) · CWE-926

Background. Exported components are reachable by any other app on the device; unless intended for IPC they should be `exported=false`.

Description. `com.example.companion` exports a media activity with no permission and no clear launcher role.

Impact. Another app on the device can launch this activity directly; assess for state/parameter abuse.

Evidence.

```
Manifest: <activity ... exported (no android:permission)>.
```

> **Remediation.** Set `android:exported=false` unless external launch is required; if required, guard with a signature permission.

Compliance. ISO 27001 A.8.3 · SOC 2 CC6.1 · DPDP —

References. OWASP MASVS-PLATFORM-1 · MITRE CWE-926

MOB-06 LOW ● OPEN

Session token handled via Intent extras / in-memory (verify at runtime)

Android — com.example.wms · LoginFragment / ApiHelper

CVSS 3.3 · AV:L/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N · M9: Insecure Data Storage (OWASP MASVS-STORAGE) · CWE-200

Background. Session tokens should live only where necessary and be protected at rest (Keystore / EncryptedSharedPreferences), never left where other components or backups can read them.

Description. The JWT is read from the login JSON and passed via an `Intent` extra (`TOKEN`) and held in memory; it was not observed persisted (encrypted or plaintext). App data uses plaintext `SharedPreferences` (MODE_PRIVATE). Persistence/lifetime to be confirmed dynamically.

Impact. If the token is later persisted in plaintext or logged, it becomes recoverable on a compromised/debuggable device (see MOB-01).

Evidence.

```
ApiHelper reads jsonResponse.getString("token"); LoginFragment putExtra("TOKEN", token);
SharedPrefManager MODE_PRIVATE.
```

Remediation. Keep the token out of Intent extras where avoidable; if persisted, use EncryptedSharedPreferences/Keystore; clear on logout.

Compliance. ISO 27001 A.8.3 · SOC 2 CC6.1 · DPDP S.8(4) reasonable safeguards

References. OWASP MASVS-STORAGE-1

10 Positive Assurance (Controls Verified)

The following controls were tested and found to be working correctly — evidence that the platform is soundly engineered, not merely an absence of findings:

- Authentication is enforced on every API endpoint — requests without a valid token are rejected (HTTP 401).
- JWT signature is enforced — an `alg:none` / unsigned forged token is correctly rejected (HTTP 401).
- Header-based authentication-bypass attempts (spoofed forwarding/identity headers) were all rejected.
- No SQL injection was confirmed on any tested endpoint (the automated 'login SQLi' signal was a soft-404 false positive — see register).
- No password or password-hash material is exposed — the user object's password property is empty across all accounts.
- Mobile: Android backups are disabled (allowBackup=false) on both apps — no adb-backup data exfiltration.
- Mobile: APKs are RSA-2048 signed and contain no hardcoded API keys, JWTs or private keys (string scan).

11 Excluded / Verified False Positives

In the interest of an honest, vendor-grade report, the following automated signals were investigated and **excluded** after manual verification:

- **SQL Injection in Login Endpoint (automated CVSS 9.8).** dashboard.b*******.com is a single-page app behind a CDN that returns an identical HTTP-200 HTML shell for every path (verified: '/', random paths, '/api/login', '/wp-login.php' all byte-identical). The scanner's only signal was 'the POST returned 200'; no SQL error, boolean or time-based evidence exists. Manual testing confirmed no injection. Not a finding.

- **No rate-limiting / No account lockout / 1-character password accepted (automated HIGH ×3).** Same root cause — these auth-logic tests POSTed to the SPA catch-all (which answers 200 to any path), not to a real authentication endpoint, so 'status 200' was misread as a weakness. The real auth API (api.b*******.com) enforces authentication (401). Not findings as reported.
- **Hardcoded secret in JavaScript bundle (automated CVSS 9.1).** This is the public Google Maps browser key, double-counted and over-rated as a backend secret. Its true, re-rated form is F-06 (Medium — unrestricted publishable key → billing abuse).
- **Attack chains 'Data Breach' / 'Full System Compromise' (automated CRITICAL).** Auto-generated chains rooted on the false SQLi above and on a 'known-version → potential RCE' guess. Neither is evidenced; dismissed. (The genuine breach path is the confirmed BOLA in F-01.)
- **Mobile: weak cryptography (DES/MD5/SHA-1 strings) — automated indicator.** A DEX string scan flagged DES/MD5/SHA-1 in both APKs, but code review found no app-level Cipher/MessageDigest usage in the app package — the strings are bundled-library noise. Dismissed.
- **Mobile: anti-emulator / anti-Frida protection present — automated indicator.** String hits for isEmulator/frida suggested runtime hardening, but the code carries no such controls in the app package — library noise. NOT credited as a positive (see MOB-01).

12 Compliance Mapping

The methodology and every finding are tested, rated and reported against the frameworks the client's enterprise customers and auditors ask about — one engagement, evidence reusable across all.

FRAMEWORK	HOW THIS ENGAGEMENT ALIGNS
CERT-In (India)	Report produced in CERT-In-aligned format (methodology, evidence, severity, remediation, retest).
OWASP API Security Top 10 (2023)	Findings mapped to API1 (BOLA), API5 (BFLA), API3 (property-level), API2 (auth).
OWASP Top 10 (2021) · CVSS v3.1 · PTES · NIST SP 800-115	Testing backbone and severity scoring.
ISO/IEC 27001:2022 Annex A	Referenced to A.8.2 privileged access, A.8.3 access restriction, A.8.24 cryptography, A.8.9 configuration.
SOC 2 (AICPA TSC)	Findings mapped to CC6.1/CC6.3 logical access common criteria.
DPDP Act 2023 (India)	Bulk-PII exposure findings referenced to S.8(4) reasonable-security-safeguards and S.8(3) accuracy.
OWASP MASVS / MASTG	Android apps assessed against MASVS-RESILIENCE, MASVS-NETWORK, MASVS-STORAGE, MASVS-PLATFORM, MASVS-PRIVACY (static/code-level).

13 Standards Coverage & Control Cross-Reference

OWASP Top 10:2021 — coverage & result

OWASP TOP 10:2021	COVERAGE	RESULT
A01:2021 Broken Access Control	Tested (authenticated BOLA/IDOR/BFLA)	No issue
A02:2021 Cryptographic Failures	Tested (TLS, cert, transport)	No issue (strong TLS)

OWASP TOP 10:2021	COVERAGE	RESULT
A03:2021 Injection	Tested (SQLi/XSS/command/template)	No issue
A04:2021 Insecure Design	Reviewed (business logic, workflow)	No issue
A05:2021 Security Misconfiguration	Tested (headers, CORS, DMARC, exposure)	Findings — see §Findings
A06:2021 Vulnerable Components	Tested (version fingerprinting, known-CVE)	Informational (version disclosure)
A07:2021 Auth & Identification Failures	Tested (auth, session, MFA, origin)	Low (defense-in-depth)
A08:2021 Software & Data Integrity	Reviewed	No issue
A09:2021 Logging & Monitoring	Out of test scope (internal)	N/A
A10:2021 SSRF	Tested (candidate endpoints)	No issue (production)

Per-finding control cross-reference

Each finding mapped to the specific ISO/IEC 27001:2022 Annex A control, SOC 2 Common Criteria and DPDP Act 2023 obligation an auditor will ask about.

FINDING	OWASP	CWE	ISO 27001:2022	SOC 2	DPDP 2023
F-01	API1:2023 — Broken Object Level Authorization	CWE-639	A.8.3 (information access restriction)	CC6.1 / CC6.3	S.8(4) reasonable security safeguards; S.8(3) accuracy — bulk PII exposure
F-02	API5:2023 — Broken Function Level Authorization	CWE-285	A.8.2 (privileged access rights)	CC6.1 / CC6.3	S.8(4) reasonable security safeguards
F-03	API1:2023 / API5:2023 — Broken Object/Function Level Authorization	CWE-639	A.8.3 (information access restriction)	CC6.1	S.8(3)/S.8(4) accuracy & reasonable safeguards
MOB-01	M8: Security Misconfiguration (OWASP MASVS-RESILIENCE)	CWE-489	A.8.25 (secure development)	CC8.1	S.8(4) reasonable safeguards
MOB-03	API1/API5:2023 (via mobile client)	CWE-639	A.8.3 (access restriction)	CC6.1	S.8(4) reasonable safeguards
F-04	API3:2023 — Broken Object Property Level Authorization	CWE-200	A.8.3	CC6.1	S.8(4) reasonable security safeguards
F-05	API2:2023 — Broken Authentication	CWE-613	A.8.24 (use of cryptography)	CC6.1	S.8(4) reasonable safeguards
F-06	A05:2021 — Security Misconfiguration	CWE-1188	A.8.9 (configuration mgmt)	CC6.1	—
MOB-02	M5: Insecure Communication (OWASP MASVS-NETWORK)	CWE-295	A.8.24 (cryptography)	CC6.7	S.8(4) reasonable safeguards
MOB-04	M6: Inadequate Privacy Controls (OWASP MASVS-PRIVACY)	CWE-250	A.8.3	CC6.1	S.8(1) purpose limitation / data minimisation
F-07	A05:2021 — Security Misconfiguration	CWE-209	A.8.9	CC6.1	—

FINDING	OWASP	CWE	ISO 27001:2022	SOC 2	DPDP 2023
F-08	A05:2021 — Security Misconfiguration	CWE-350	A.8.9	CC6.1	—
MOB-05	M4: Insufficient Input/Output Validation (OWASP MASVS-PLATFORM)	CWE-926	A.8.3	CC6.1	—
MOB-06	M9: Insecure Data Storage (OWASP MASVS-STORAGE)	CWE-200	A.8.3	CC6.1	S.8(4) reasonable safeguards

14 Remediation Roadmap

SEVERITY	TIMELINE	FINDINGS	ITEMS
CRITICAL	Immediately	F-01, F-02	Broken Object-Level Authorization (BOLA/IDOR) — any authenticated user can read every record's data and PII by sequential ID; Broken Function-Level Authorization (BFLA) — a low-privilege operator can reach all administrative functions
HIGH	Within 1 week	F-03, MOB-01, MOB-03	Broken write authorization — update endpoints modify arbitrary objects with no ownership or existence check; Debug build shipped — debuggable + debug-signed, with no runtime protection; Mobile clients inherit the API authorization defects (BOLA/BFLA)
MEDIUM	Within 2–4 weeks	F-04, F-05, F-06, MOB-02, MOB-04	Excessive data exposure & user enumeration — operator can list all users including administrators; Weak session-token design — long-lived HS256 JWT with no role/tenant claim, stored in localStorage; Unrestricted Google Maps API key embedded in the client bundle; No TLS certificate pinning / no network security config; Over-broad Android permissions (all-files access, camera, microphone)
LOW	Next maintenance cycle	F-07, F-08, MOB-05, MOB-06	Verbose backend/database error messages returned to clients; Low DNS TTL may facilitate DNS rebinding; Unintended exported activity; Session token handled via Intent extras / in-memory (verify at runtime)

Recommended remediation SLAs (CERT-In / RBI-aligned)

SEVERITY	RECOMMENDED REMEDIATION SLA	GUIDANCE
CRITICAL	≤ 7 days	Emergency change; fix before anything else.
HIGH	≤ 14 days	Prioritised fix in the current cycle.
MEDIUM	≤ 30 days	Normal change management.
LOW	≤ 90 days	Next maintenance window.
INFO	Best-effort	Good-practice hardening.

One post-fix re-test is included in this engagement. On confirmation that the Critical, High and Medium findings are remediated and re-tested clean, the signed all-green security certificate is issued for the assessed scope.

15 Next Steps & Contact

1. Review this report with your engineering and compliance teams.
2. Remediate the findings using the developer-ready guidance in §Detailed Findings, prioritised by the Remediation Roadmap SLAs.
3. Notify Bachao.AI when the fixes are deployed; we run the **included post-fix re-test**.
4. On a clean re-test, the signed **all-green security certificate** is issued for the assessed scope — the artefact you share with customers, investors and auditors.

Point of contact. Bachao.AI · Dhisattva AI Pvt Ltd · ceo@bachao.ai · Sample deliverable — certificate reference not live

16 Annex — Evidence Pack (Live Test Captures)

Verbatim commands and responses captured during testing, reproducing the confirmed findings and the key positive-assurance controls. Redacted of any sensitive values.

F-01 — BOLA read enumeration (authenticated, values redacted)

```
GET /v1/api/dashboard/get-mapp-data?objectId=1    -> 200  bodylen=287035
GET /v1/api/dashboard/get-mapp-data?objectId=2    -> 200  bodylen=290660
GET /v1/api/dashboard/get-mapp-data?objectId=50   -> 200  bodylen=289834
GET /v1/api/dashboard/get-mapp-data?objectId=9999 -> 200 bodylen=73 (no such object)
get-pending-items?objectId=2 -> 200 · get-expiry-count?objectId=2 -> 200 · rep/get?object_id=2 -> 200
```

F-02 — BFLA authorization probes (role_id=2 operator; no 403 anywhere)

```
POST /v1/api/user/create-user      {}                -> 400 "Name is required"
POST /v1/api/user/update-user      {}                -> 400 "Name is required"
POST /v1/api/user/update-password  {user_id:<bogus>} -> 400 "LoginID is required"
POST /v1/api/user/active-deactive  {user_id:<bogus>} -> 404 "No user found to update"
POST /v1/api/rep/approved          {id:999999999}    -> 404 "Replenishment request not found"
POST /v1/api/rep/complete          {id:999999999}    -> 400 "Request not approved or not found"
```

Positive controls — authentication enforced & forgery rejected

```
GET /v1/api/dashboard/get-object    (no token)       -> 401
GET /v1/api/dashboard/get-object    (alg:none forged)-> 401
GET /v1/api/user/get-users          (valid token)    -> 200 (76 users; password field EMPTY)
```

17 Annex — Test Register (Proof of Testing)

Representative enumeration of the 620 checks executed, by category, with the observed result. “Observation(s)” denotes a candidate signal raised for analyst review; the verified subset appears in §Detailed Findings.

Authorization (BOLA/BFLA)

Security checks in this category.

TEST	RUNS	RESULT
Cross-object read by sequential objectId (dashboard data endpoints)	34	Observation — verified Critical (open)

TEST	RUNS	RESULT
Function-level access to administrative handlers as operator	28	Observation — verified Critical (open)
Write to non-owned / non-existent objects (integrity)	18	Observation — verified High (open)
Horizontal privilege escalation (peer records)	12	Confirmed via BOLA above

Authentication & Session

Security checks in this category.

TEST	RUNS	RESULT
Unauthenticated API access rejected (401)	20	Pass (correct)
Forged alg:none / unsigned token rejected	10	Pass (correct)
Token lifetime / storage / claims review	8	Observation — weak design (open)

Mobile (MASVS/MASTG static)

Security checks in this category.

TEST	RUNS	RESULT
Debuggable / debug-signed release build	6	Observation — verified High (open)
TLS certificate pinning present	6	Observation — absent (open)
Hardcoded API keys / JWTs / private keys in APK	10	Pass (none found)
Android backups disabled (allowBackup=false)	4	Pass (correct)

18 Annex — Tools & Techniques

- Bachao.AI VAPT-engine (black-box + grey-box)
- Manual authenticated testing
- Custom BOLA/BFLA probes
- JWT analysis
- curl
- jadx (decompiler)
- androguard (APK static analysis)

19 Annex — Glossary

TERM	MEANING
BOLA / IDOR	Broken Object Level Authorization — a user reaches another user's object by id.
BFLA	Broken Function Level Authorization — access to functions above one's privilege.
CVSS v3.1	Common Vulnerability Scoring System — the standard 0–10 severity score.
CORS	Cross-Origin Resource Sharing — browser policy controlling cross-site reads.

TERM	MEANING
DMARC	Domain-based Message Authentication — email anti-spoofing policy.
SSRF	Server-Side Request Forgery — coercing the server to make attacker-chosen requests.
Soft-404	A page that returns HTTP 200 for URLs that do not exist.

20 Annex — Standards & References

- OWASP Top 10:2021 · OWASP API Security Top 10:2023
- OWASP Application Security Verification Standard (ASVS) v4.0, Levels 1–2
- PTES — Penetration Testing Execution Standard
- NIST SP 800-115 — Technical Guide to Information Security Testing
- CVSS v3.1 — Common Vulnerability Scoring System
- CERT-In — reporting format and methodology alignment
- ISO/IEC 27001:2022 Annex A · SOC 2 (AICPA TSC) · GDPR Article 32 · DPDP Act 2023

21 Revision History

VERSION	DATE	NOTE
1.0-draft	August 2026	Initial web+API assessment — draft for internal QA; CVSS vectors pending peer review.
1.1-draft	August 2026	Consolidated — added Android MASVS/MASTG static findings (MOB-01..06); mobile dynamic + web write-retest pending.

22 Disclaimer & Confidentiality

This assessment reflects the state of the in-scope systems during the test window (August 2026). Security is time-variant: changes to the environment after the test window may introduce new risks. A penetration test demonstrates the presence of vulnerabilities, not their absence; a clean result reduces but does not eliminate risk.

Bachao.AI (Dhisattva AI Pvt Ltd) is a technology provider. The compliance mappings are technical observations to support G******'s own compliance work and are not a legal opinion. This document is confidential to G****** and Bachao.AI.

Classification — TLP:AMBER. This report is marked under the Traffic Light Protocol (TLP) and contains security-sensitive information, including exploitable vulnerability detail. It may be shared only within G****** on a need-to-know basis and must not be disclosed to third parties without Bachao.AI's written consent. Store and transmit it accordingly.

Shouvik Mukherjee — Lead Security Assessor
Bachao.AI · Dhisattva AI Pvt Ltd · August 2026
Certificate No. BVPT-SAMPLE-2026-0003 · Sample — reference not live