



CONSULTING SERVICES PVT. LTD.

ETHOS TERMINAL SERIES · PRODUCT DATASHEET · DOC ETH-DS-AT01 · REV 1.0 · AUGUST 2026

ethos.

Smart Attendance Terminal

MODEL ETH-AT01

Look at the camera — you're marked in. Contactless, shift-aware attendance with on-device face recognition, fingerprint and RFID fallback, and serverless multi-terminal sync over your LAN.

Three ways to punch

Face recognition · fingerprint (10 prints/user) · RFID card (optional)

128-D face engine, on device

No cloud calls — detection, landmarks & matching run locally

Shift-aware records

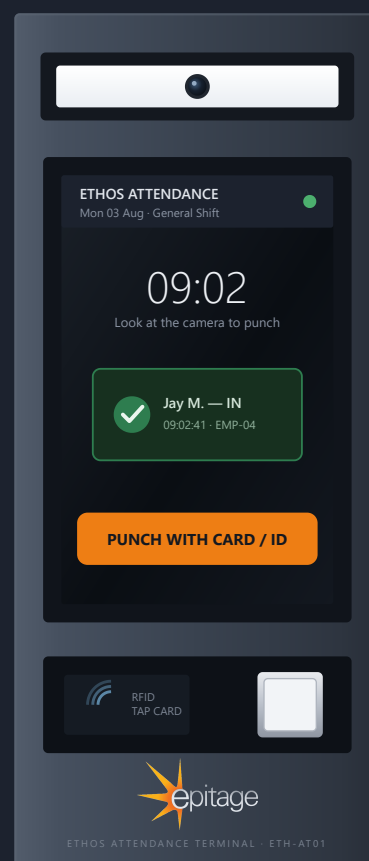
Every punch carries shift & time-window context automatically

Instant feedback

Green LED, on-screen confirmation and voice prompt in under a second

Multi-terminal mesh

Enrolments replicate peer-to-peer over UDP — no central server needed



1 Product overview

The **Ethos Smart Attendance Terminal (ETH-AT01)** turns walking past a wall into marking attendance. The illuminated camera bar recognises a registered face in under a second of camera time, stamps the punch with the **correct shift and time-window context**, confirms on screen with a green LED and a voice prompt — no cards to forget, no queues at a register, nothing to touch.

Every terminal is self-contained: the kiosk UI, face matching and the attendance database all run on the device. Multiple terminals across gates, floors or buildings form a **peer-to-peer mesh** on your LAN — enrol a person once at any terminal and every other terminal recognises them — with no central server to install, license or maintain.

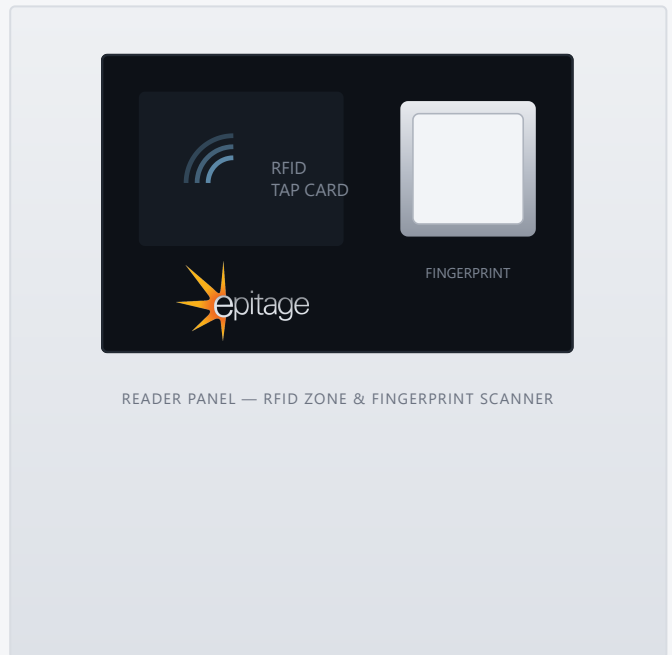
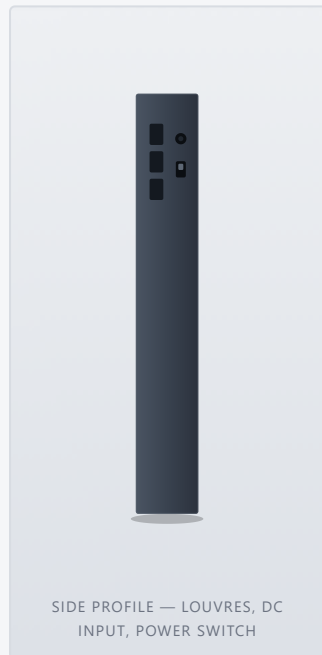
ETH-AT01 shares its platform with the Ethos Smart Canteen Terminal (ETH-CT01) — the same design language, electronics and software base in a shorter enclosure without the printer bay and meal modules — so a mixed fleet behaves, enrolls and administers identically.

At a glance

Identification	Face recognition (primary) · Fingerprint, 10 prints/user · RFID card †
Face engine	128-D descriptors · match < 0.4 Euclidean · ~300 ms scan loop
Display	7-inch capacitive touch panel †
Punch record	employee · shift · time window · IST timestamp · via shared transaction channel
Compute	Raspberry Pi-class Linux SBC †
Connectivity	Ethernet / Wi-Fi LAN · REST :5000 · UDP mesh :41234
Audio	Voice prompts through built-in speaker
Status LED	4-state indicator (idle / scanning / pass / fail)
Power	12 V DC barrel jack · rocker power switch †
Mounting	Rear keyhole slots — wall mount, tool-free lift-off

† Indicative — final values pending production drawings.

2 Product views



Rendered product illustrations are representative; final finish and dimensions per production drawings.

3 Where it fits

Factory gates & floors

Shift-based workforces around the clock — the seeded configuration covers 5 shifts and 8 time windows including overnight shifts, so every punch lands in the right context.

Offices

Contactless walk-in attendance at reception or per floor; multiple terminals stay in sync over the LAN without an attendance server.

Campuses & contractors

Fast enrolment at any terminal, instant recognition at all of them — suited to sites where the workforce changes often.

4 Hardware anatomy



#	ELEMENT	DETAIL
1	Illuminated camera bar	USB camera behind a diffused white light strip — consistent face illumination indoors, day or night. Captures at 320 × 240 for the recognition pipeline.
2	Touch display	Portrait-mounted capacitive panel running the kiosk UI in a full-screen browser. On-screen keyboard for operator entry — no physical keyboard needed.
3	RFID reader zone	Tap area for card-based punches as a fallback or companion to face ID. [†]
4	Fingerprint scanner	Optical scanner with white surround illumination. Enrolment stores up to 10 fingerprints per user (all fingers, both hands).
5	Status LED & speaker	Side-mounted indicator LED plus internal speaker for voice prompts; louvres double as ventilation.
6	Branding plate	Customisable fascia — your company branding on request.

Status LED language

- ☐ **White, steady** — idle, ready for the next person
- ☐ **White, blinking** — scanning fingerprint / capturing
- ☒ **Green** — punch recorded
- ☐ **Red** — authentication failed

Voice prompts

“Thank You”	Punch recorded successfully
“User Already Taken”	Already punched for this window
“No User Found”	Person not registered on this terminal

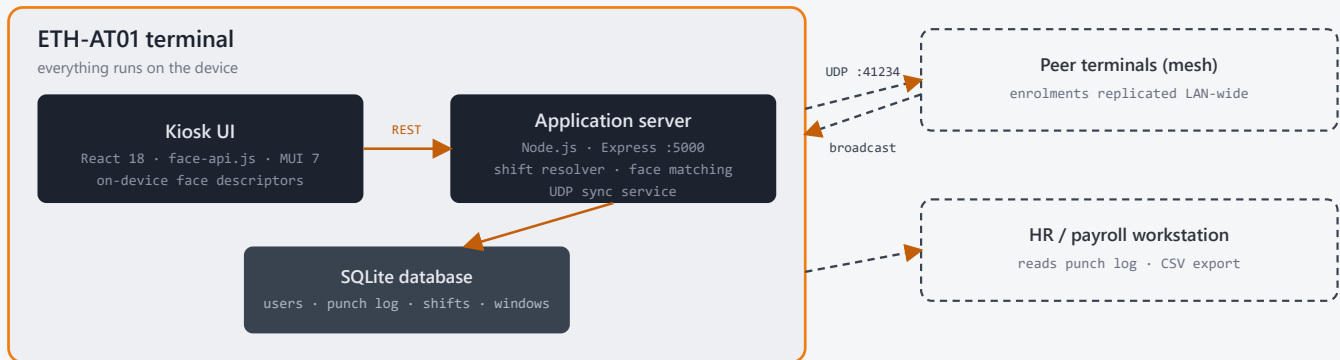
5 Physical & electrical

Enclosure	Powder-coated sheet-steel tower, graphite grey; crisp folded fascia
Dimensions (H×W×D)	≈ 370 × 150 × 90 mm [†]
Weight	≈ 2.5 kg [†]
Mounting	Two rear keyhole slots on vertical centre-line; lift-off removal
Recommended height	Camera bar at 1.4 – 1.6 m from floor

[†] Indicative — final values pending production drawings.

Input power	12 V DC barrel jack, external adapter [†]
Power switch	Side rocker switch
Operating environment	Indoor, dry locations; keep direct sunlight off the camera bar
Cooling	Passive, side louvres
Consumables	None — no printer, no paper

6 System architecture



7 Software stack

LAYER	TECHNOLOGY
Kiosk UI	React 18.2 · MUI 7 · react-simple-keyboard 3.8
Face engine	face-api.js 0.22.2 (TensorFlow.js, in-browser)
API server	Node.js · Express 4.21 · CORS enabled
Database	SQLite 3 (sqlite3 5.1) — single file, zero admin
Sync	Node dgram — JSON datagrams, UDP broadcast :41234
Delivery	Kiosk browser in full-screen on the terminal itself

Attendance profile: ETH-AT01 runs the shared Ethos software base with the meal-selection, quota and printing modules switched off — identification itself is the punch; punch persistence and sync use the platform's shared transaction channel, enabled during site commissioning.

8 Face recognition engine

Detector	TinyFaceDetector · input 160 px · score ≥ 0.5
Landmarks	68-point model
Descriptor	128-dimension float vector per face
Matching	Nearest neighbour, Euclidean distance < 0.4
Scan cadence	Live detection every 300 ms on 320x240 video
Enrolment	Single capture per user, operator-assisted, on the terminal
Privacy	Only the numeric descriptor is stored — no face photos are saved
Locality	All models load from the device; recognition works with no internet

9 Terminal REST interface

ENDPOINT	METHOD	FUNCTION
/api/identify-user	POST	Match a face descriptor; resolves the employee plus the current shift and time window and returns duplicate and previous-shift flags. Punch persistence uses the platform's shared transaction channel, enabled during site commissioning.
/api/register-user	POST	Enrol name + employee ID + face descriptor; rejects a face already enrolled; replicates the new user to all terminals.
/api/employees	GET	Employee directory (ID + name) for the operator search on the registration screen.

10 A punch in five steps



If the person is unknown or already punched — clear on-screen notice, voice prompt, red LED, and duplicate flags are raised instead.

11 Shift-aware by design

ETH-AT01 does more than timestamp: every punch is resolved against the site's **shift and time-window configuration**, held as plain data in the terminal database:

1 · Window	The wall clock resolves the active time window from <code>time_slot_master</code> — overnight windows are handled correctly (e.g. 22:00 → 04:00 belongs to the night shift).
2 · Shift	<code>shift_time_slot_mapping</code> links the window to its shift, so the punch carries shift context without the employee choosing anything.
3 · Duplicates	An existing punch in this window raises a duplicate flag at identification time; a previous-shift check flags anyone reappearing under a different shift the same day.
4 · Replication	Enrolments broadcast to every peer terminal automatically; punch records replicate through the shared transaction channel.

Factory-seeded configuration

SHIFTS (5)

Morning 06:00–14:00
General 09:00–17:00
Split 10:00–14:00
Afternoon 14:00–22:00
Night 22:00–06:00

TIME WINDOWS (8)

F1 07–09 · F2 10–13
G1 09–10 · G2 10–14
S1 14–17 · S2 18–22
T1 22–04 · T2 03–05

Shifts and windows are data, not code — reshape them per site (HR calendars, seasonal hours) without touching the application. The factory seed maps no windows to the Split Shift — map or remove it per site.

Reporting

Punches persist in the platform's shared transaction table (`order_table`) with employee, shift, window and IST timestamp — one query away from an attendance register:

```
-- who punched today, by shift
SELECT u.name, o.shift_id,
       MIN(o.timestamp) AS first_in
FROM order_table o JOIN users u
  ON u.empId = o.user_id
WHERE DATE(o.timestamp) = DATE('now','localtime')
GROUP BY u.name, o.shift_id;
```

Read the SQLite file directly, or export CSV from any SQLite tool for payroll.

12 Enrolling people

01 Open registration

Tap **Register User** on the home screen.

02 Find the employee

Search by name or ID with the on-screen keyboard — the directory suggests the top matches; tap one to fill the form.

03 Capture the face

Position the person 40–60 cm away facing the camera; capture once the detection box appears and the hint flips to *"Face detected!"*. The terminal computes the 128-number descriptor, stores it, and replicates it to every peer terminal.

04 Fingerprints (optional)

Enrol up to 10 prints per user — all fingers of both hands — for glove-free fallback punching.

Good captures: even, front-on lighting (the camera bar provides it), glasses off if they glare, one face in frame, neutral expression. Re-enrol after major appearance changes.

One face, one ID: a face already on file is rejected at enrolment with the existing employee named — one face can never map to two IDs, on any terminal in the mesh.

13 Commissioning guide

Before you start: terminal mounted with the camera bar at 1.4–1.6 m · 12 V supply connected · terminal joined to the site LAN. Software prerequisites on the terminal: Node.js LTS with npm, and a Chromium-family kiosk browser.

01 Install the application server

Copy the server/ package to the terminal and install dependencies:

```
cd Canteen/server
npm install
npm start      # "Server running on port 5000"
```

Always launch from inside `server/` — the database path is resolved from the working directory. The shared software package ships in the `Canteen/` directory.

02 Verify the database

The SQLite database must sit beside `server.js`, pre-seeded with your shifts, time windows and their mappings (section 11). The server does not create the schema on its own.

03 Build the kiosk UI

```
cd Canteen/client
npm install
npm run build      # production bundle in client/build/
```

Serve `client/build/` from any static server on the terminal and open it in the kiosk browser, full-screen. The face models ship with the build and load locally at boot.

04 Grant the camera

On first launch the browser asks for camera access — allow it, and set the kiosk browser to remember the grant. Camera capture requires the UI to be served from `localhost` or `HTTPS`.

05 Join terminals into a mesh (optional)

Put every terminal on the same LAN broadcast domain and allow UDP 41234 between them — that is all: discovery is automatic, peers replicate enrolments and punches, and a terminal ignores its own broadcasts.

06 Reprint the UI at a remote server (optional)

The kiosk UI ships pinned to `http://localhost:5000`. To run the UI and server on different hosts, update the URL at its call sites and rebuild:

```
# src/components/faceCapture.jsx      (identify)
# src/components/faceRegistration.jsx (employees, register)
npm run build
```

07 Configure shifts & windows

Adjust the shift and window tables to your site (via SQL today — Time & Shift Management admin modules per the platform specification are in development):

```
-- a new window, tied to General Shift (id 2)
INSERT INTO time_slot_master
(TimeSlotName, start_time, end_time)
VALUES ('Evening window', '17:00', '19:00');
INSERT INTO shift_time_slot_mapping VALUES (2, 9);
```

08 Smoke-test

Register a test user (section 12), punch them in, and confirm: green LED, "Thank You" prompt, and the punch visible in the database — then delete the test rows.

Cover the clock: when a moment of the day falls outside every configured window, a punch has no shift context — keep windows contiguous across your working hours (the factory seed leaves 05:00–07:00 and 17:00–18:00 open by design).

14 Care & maintenance

Daily

Wipe the screen and camera window with a dry microfibre cloth. Glance at the status LED — steady white means ready.

Monthly

Dust the side louvres. Verify terminal clock against IST — all shift decisions are wall-clock based. Back up the database (a simple file copy after hours).

Per season

Review shift & window tables against HR calendars. Re-enrol faces flagged by staff as slow to match. Archive old punch rows to keep queries brisk.

15 Limitations & deployment guidelines

IDENTIFICATION		PLATFORM & NETWORK	
Indoor use	Camera pipeline is tuned for indoor light; avoid direct sun or strong backlight on the approach.	LAN product	REST and UDP sync are open on the LAN with no authentication — deploy on an isolated/segmented VLAN only.
One face at a time	The first detected face is processed — queue people single-file at the terminal.	Biometric transport	Face descriptors travel over plain HTTP on the local link (and appear in the browser debug console in this build) — keep the kiosk network isolated; TLS is planned.
No liveness check	Descriptor matching does not detect photos; supervise enrolment and place terminals in staffed or camera-covered areas.	Mesh semantics	UDP replication is fire-and-forget: no delivery guarantee or conflict resolution. Peers that were offline do not back-fill missed punches.
Open enrolment	Registration is reachable from the idle screen and employee details are operator-typed — restrict enrolment to supervised sessions until operator login ships.	IN/OUT semantics	The terminal records punch events; pairing them into IN/OUT and durations is done at reporting time (or via site customisation).
Single enrolment	One face capture per user; large appearance changes may need re-enrolment.	Time & locale	IST (Asia/Kolkata) fixed; correct system time is essential — punches inherit the terminal clock.
Population size	Matching scans all users per punch — sized for workforces in the hundreds; contact us beyond ~1,000.	Server address	UI is compiled against localhost:5000 — remote-server layouts need a rebuild (\$13.6).

16 Known issues — software build under review

ISSUE	IMPACT & PLANNED FIX
Cross-shift duplicate check date handling	On the 1st–10th of a month the previous-shift window can be mis-computed (unpadded dates), weakening the cross-shift flag on those days.
Peer “identified” status not applied	The status-sync message references a column absent from the live schema, so the status-sync path is inoperative in this build; identification itself is unaffected.
On-screen keyboard stays open	On the registration screen the keyboard cannot be dismissed until the camera step.
Camera stream lingers	The webcam can remain active after leaving the identify/registration screens until the flow resets.

Issues are listed for transparency toward pilot deployments; none block the core identify → punch loop.

17 Troubleshooting

SYMPTOM	LIKELY CAUSE	ACTION
"Failed to load face recognition models"	Model files missing from public/models/	Restore the three model sets (detector, landmarks, recognition) and reload the kiosk page.
Camera never starts	Permission denied, or page not on localhost/HTTPS	Re-grant camera permission in the kiosk browser; serve the UI from localhost or HTTPS.
"User not recognized" for enrolled staff	Lighting change, appearance change, or match > 0.4	Retry facing the camera bar squarely; re-enrol the user if it persists.
Punch has no shift context	Clock outside every configured window, or shift unmapped	Check terminal time; verify the current time falls in a window mapped to a shift (\$13.7).
Punches not appearing on other terminals	Different subnet or UDP 41234 blocked	Put terminals in one broadcast domain; allow UDP 41234 in switch/firewall policy.
All users unrecognised / employee list empty	Started outside server/ — a fresh empty DB was created	Stop, delete the stray database file, start again from the server directory.

Ordering & variants

ETH-AT01	Smart Attendance Terminal — this document
ETH-CT01	Smart Canteen Terminal — same platform with 58 mm token printer and meal rule engine (separate datasheet ETH-DS-CT01)
Options	Branding fascia · languages · enclosure colour · IN/OUT pairing & payroll integration · site-specific rules

Support

Epitage Consulting Services Pvt. Ltd.

enquiry@epitage.com

+91 98503 01345

Sales, commissioning, site customisation and AMC support for the entire Ethos Terminal Series.

© 2026 Epitage Consulting Services Pvt. Ltd. Specifications describe the software build reviewed in August 2026 and are subject to change; values marked † are indicative pending production drawings. This document does not constitute a warranty.