Akash Yadav, 2026, 14:6
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

AI-Powered Employee Performance and Activity Tracking System

Research Scholar Akash Yadav, Shashank Pandey, Assistant Professor Mr. Sachin Chauhan
CSE Department HMRITM GGSIPU, Delhi

Abstract- Electronic monitoring of employees has expanded with remote and hybrid work, yet meta-analytic evidence shows that monitoring does not, on average, improve performance and is associated with higher stress, while monitoring systems routinely treat activity signals such as keyboard and mouse events as if they measured productive work. This paper reports the design, implementation and evaluation of WorkPlus, an open-source employee performance and activity tracking system built as a design science artefact. The system combines (i) consent-gated, content-free activity measurement by a cross-platform desktop agent, (ii) task contextualisation through a manager-reviewed task lifecycle with due dates and expected effort, and (iii) AI-assisted reporting in which the server computes every number from records, a large language model (LLM) only interprets a numbered fact sheet, and a grounding check removes any statement whose numbers are not supported by cited facts. The implementation comprises a React client, a Node.js/Express API with MongoDB, a Python statistical service, an Electron agent and two optional LLM-based agents. The authors evaluated the system on a seeded synthetic dataset (four employees, 14 days). All 190 automated tests (242 test points including subtests) across five components passed; the project's evaluation harness passed 88 of 88 hand-computed metric and data-quality checks and 10 of 10 authorisation cases; 23 black-box API tests passed; and median latency of authenticated read endpoints was 5.0–17.3 ms on a two-vCPU host with a local database. An adversarial perturbation test of the grounding check rejected all statements with missing, invalid or swapped citations and 92% of statements with an altered number, with no false rejections of 104 valid statements, but it accepted every wrong number written as a word and every unsupported qualitative judgement, and numbers that happen to match the day or month of a cited date. Inspection of eight generated reports exposed two defects: non-working days enter the trend and anomaly series as zero-score days, and the statistical service and dashboard aggregate scores differently. The LLM path was not evaluated because no API key was used, and no study with real employees was conducted. The contribution is an evidence-linked, human-supervised reporting architecture that keeps activity, task outcomes and inferred performance separate, together with an evaluation that identifies its current limits.

Keywords- Electronic performance monitoring; algorithmic management; privacy-aware design; task contextualization; large language models; grounding; explainable AI; human–AI interaction; design science research; workforce analytics.

I. INTRODUCTION

1. Background

The organisation of knowledge work changed markedly after 2020. A natural-experiment study of 61,182 Microsoft employees found that firm-wide remote work made collaboration networks more static and siloed and shifted communication from synchronous to asynchronous channels [1]. When managers can no longer observe work directly, they increasingly rely on software to make work visible. This shift is part of a broader movement towards algorithmic control, which Kellogg, Valentine and Christin characterise through six mechanisms: restricting, recommending, recording, rating, replacing and rewarding [2]. Employee activity trackers belong mainly to the "recording" and "rating" mechanisms: they capture behavioural traces and convert them into scores.

The technical barrier to such systems is low. Operating systems expose idle time, input events and the foreground application to any installed agent, and cloud LLMs can turn tabular data into fluent narratives within seconds. The more difficult

© 2026 Akash Yadav. his is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly credited.

questions are therefore not whether activity can be measured, but what the measurements mean, who may see them, and how automated interpretations can be kept accountable to the underlying records. This paper addresses those questions through an implemented and publicly deployed system.

II. PROBLEM STATEMENT

Two problems motivate this study. The first is a validity problem. Commercial and academic trackers commonly present activity indicators (active time, keystroke counts, application usage) as measures of productivity. A meta-analysis of 70 samples found that electronic monitoring had no relationship with performance ($r = -.01$) but small adverse associations with job satisfaction, stress and counterproductive work behaviour [4]. An activity signal is at best evidence of presence at a device; treating it as output invites both misjudgement and gaming.

The second is a governance problem. A European Commission Joint Research Centre review of 398 studies links excessive monitoring to resistance, stress and turnover intentions, and recommends policy grounded in privacy, data justice and organisational justice [3]. When AI is added, a third risk appears: an LLM asked to "summarise performance" can produce confident statements that the data do not support. The research problem is therefore: how can an employee tracking system measure workplace activity, place it in the context of assigned tasks, and generate AI-assisted reports, while keeping activity separate from performance judgements, collecting only consented content-free data, and making AI statements checkable against records?

Research Objectives

König's review argues that the effects of monitoring depend on how it is decided, introduced and continued, and calls for designs that preserve worker agency and transparency [11]. Consistent with that position, the study pursued five objectives:

- Design and implement a multi-role system (administrator, manager, employee) that records consented, content-free activity signals and task events.
- Contextualise activity with task data (assignment, due date, expected effort, submission and approval) so that delivery can be reported separately from presence.
- Integrate AI-assisted reporting in which computation is deterministic and the AI's role is limited to interpretation that can be checked against evidence.
- Enforce privacy controls (consent per data type, pause/resume, scoped access, retention limits, view auditing) on the server rather than only in the interface.
- Evaluate functional correctness, metric correctness, access control, robustness to faulty data and performance with reproducible tests, and report the limits of that evidence.

Research Questions

- **RQ1.** Can activity measurement and task contextualisation be combined in one architecture that keeps observable activity, task outcomes and inferred performance as distinct, separately labelled constructs?
- **RQ2.** Which classes of unsupported AI-assisted statements can an automatic evidence-grounding check detect, which can it not, and how does the system behave when the AI is unavailable?
- **RQ3.** Which privacy and access-control properties can be enforced and verified at the server level, and which remain dependent on organisational practice?
- **RQ4.** Does the implementation meet its functional, correctness and performance requirements under controlled test conditions, and what defects or threats to validity does the evaluation reveal?

Scope of the Study

The study covers the WorkPlus repository at commit d3ee6ef (11 November 2026) and its public deployment (client on Vercel, API on Render, database on MongoDB Atlas). In scope are the web client, REST API, database schema, desktop agent, statistical AI service, the server-side AI assistant, and the two optional LLM agents (screenshot analysis

and camera labelling) as far as their code and unit tests allow. Out of scope are a field deployment with real employees; evaluation of LLM output quality with a live model; usability testing; formal legal compliance assessment; and load testing beyond a single small server. All quantitative results come from synthetic data generated by the project's deterministic seed script, and no personal data of real employees was processed.

Research Contributions

The study does not claim a new machine-learning algorithm. The AI service uses an existing weighted average, ordinary least-squares slope and z-scores, and the assistant uses an externally hosted LLM. The contributions are architectural and evaluative, and each is tied to the evidence that supports it in Table 1.1, in line with the design science requirement that artefact claims be evaluated [21].

- **C1. Separation of constructs.** Activity, task outcomes and inferred indicators are stored, computed and labelled separately; every presence fact sent to the assistant carries the label "Recorded presence, not a measure of productive work."
- **C2. Evidence-grounded AI-assisted reporting.** A pipeline in which the backend calculates, the model only interprets, and the server verifies each statement with a numeric grounding check; a rule-based fallback passes through the same check.
- **C3. Server-enforced privacy controls.** Per-type consent, tracking state with a 90-second grace window, team scoping, token revocation on sign-out or password change, screenshot view logging and TTL-based retention, enforced in the API rather than the user interface.
- **C4. An evaluation package** (seeded dataset, scenario harness, benchmark and perturbation scripts supplied with this paper) and an evaluation that reports discovered defects and the evidence that is still missing.

Table 1: Contribution-to-evidence matrix

Contribution	Problem addressed	How implemented	Evidence in this study	Not yet validated
C1 Separation of constructs	Activity treated as productivity [4], [13]	Separate collections (Activity, Task, Performance); presence labels in agentFacts.js; task metrics judged by submission time	Scenario S11 (presence label), S12 (on-time by submission); F20 black-box check	Whether managers actually interpret the labels correctly (needs user study)
C2 Grounded AI reporting	Unsupported or hallucinated AI statements [23]	agentFacts → LLM (structured output) → agentGrounding.groundOutput; ruleInsights fallback	Perturbation test: 100% rejection of missing/invalid/swapped citations, 92% of altered numbers, 0 false rejections (n = 104); grounding unit tests	Numbers as words and qualitative claims not detected; LLM groundedness and injection resistance (harness Part C not run)

Contribution	Problem addressed	How implemented	Evidence in this study	Not yet validated
C3 Server-enforced privacy	Covert or excessive monitoring [3], [11]	Consent checks in controllers; tracking state; roleMiddleware scoping; tokenVersion and session checks; TTL indexes; views[]	Server tests incl. screenshot-scope API; F05–F11, F22; paused/duplicate/overlap ingestion tests	Employee comprehension of consent; organisational misuse outside the software
C4 Evaluation package	Unverifiable claims in tracker studies	Seeded dataset; evaluation/runEvaluation.js (repository); bench.mjs, perturb.mjs, e6/ablation scripts (supplied with the test report)	190 tests / 242 test points, 88/88 checks, 10/10 cases, 23/23 API checks, latency/throughput, 8-report inspection	Seed output depends on run date; results on real workforce data; external replication

III. LITERATURE REVIEW

The review is a targeted narrative review, not a systematic review, and is not PRISMA-compliant. Sources were identified in Google Scholar, publisher databases (ACM Digital Library, SpringerLink, ScienceDirect, Wiley, SAGE, IEEE Xplore records) and official legal portals, using combinations of the terms electronic performance monitoring, algorithmic management, employee surveillance, AI in HRM, developer productivity, explainable AI, human–AI interaction, time-series anomaly detection, LLM hallucination and attribution. Peer-reviewed work from 2018–2026 was prioritised, with older foundational methodology retained; bibliographic details were verified on 22 November 2026. Commercial monitoring products were not systematically reviewed, which is a limitation of this section.

Existing Employee Performance Tracking Systems

Recent conference work presents AI as a remedy for the weaknesses of traditional appraisal. Madhumita et al. argue conceptually that analytics, machine learning and natural-language processing can replace infrequent, subjective reviews with continuous monitoring, prediction and personalised feedback [5]; no empirical sample is reported in the abstract. Suvedhaa et al.'s paper on AI-powered workforce management for automated appraisals and task allocation [6] points in the same direction; its full text could not be obtained, so it is cited here only for its stated topic and is not used in the comparisons below. These works illustrate a design pattern in which automation is applied to the evaluative judgement itself.

Research on algorithmic management questions that pattern. Duggan et al. show how app-based platforms use algorithms for work assignment and performance management, reshaping employment relations [7]. Jarrahi et al. argue that as these practices spread from gig platforms into conventional organisations, they shift power between workers and managers and create information asymmetries, with opacity limiting what workers can know about how they are assessed [8]. Kellogg et al. add that workers respond with "algoactivism", contesting or gaming the metrics [2]. The common lesson is that a tracking system is not neutral infrastructure: what it records, and who can see the logic, shapes work.

Employee Activity Monitoring

Ravid et al. provide an integrative review of electronic performance monitoring (EPM), proposing a typology of monitoring characteristics—such as purpose, invasiveness and transparency—and a research agenda [9]. Their later meta-analysis of 94 independent samples (N = 23,461) found no evidence that monitoring improves performance, found that it is associated with higher

stress regardless of configuration, and found that more transparent and less invasive monitoring is associated with better attitudes [10]. Siegel et al. reach a consistent conclusion with a different sample of studies [4], and König's 2025 review describes the overall effect on performance as roughly neutral, with small positive associations with strain and small negative associations with job attitudes [11].

Two points follow for system design. First, if monitoring does not raise performance on average, a system cannot justify itself by monitoring more; it must justify itself by making work more understandable and support timelier. Second, the moderating role of transparency and invasiveness [10] turns specific design choices—content-free counts rather than keystroke content, opt-in screenshots, visible tracking state—into variables that matter. Ball's policy review reaches the same conclusion from a regulatory angle, recommending data minimisation and worker participation [3]. The remote-work evidence [1] explains why demand for such tools persists despite these findings.

AI-Based Performance Analysis

Tambe, Cappelli and Yakubovich identify four obstacles to AI in human resources: the complexity of HR outcomes, small datasets, fairness and legal accountability, and negative employee reactions [12]. "Performance" in knowledge work is a construct without a single ground truth, which makes supervised learning of performance labels hard to justify. Forsgren et al. make the same point for software development: productivity cannot be captured by any single metric, and their SPACE framework recommends combining measures from at least three of five dimensions (satisfaction and well-being, performance, activity, communication and collaboration, efficiency and flow), including at least one perceptual measure [13]. Activity is one dimension among five, not a proxy for the whole.

Explainability is the second theme. Barredo Arrieta et al. survey XAI concepts and taxonomies and link explainability to fairness and accountability within "responsible AI" [14]. For HR applications, transparent-by-design models—where the computation is inspectable—avoid the need to explain a black box after the fact. Parent-Rocheleau and Parker show that algorithmic management functions such as monitoring, goal setting and performance management change job demands and resources, with transparency, fairness and human influence as moderators [15]. Taken together, these works suggest that AI-based performance analysis should use inspectable computations, multiple dimensions and human decision rights.

Intelligent Agents and AI-Assisted Analytics

LLMs offer a new interface to workplace data, but their benefits are uneven. In a preregistered field experiment with 758 consultants, Dell'Acqua et al. found that GPT-4 users completed more tasks, faster and at higher quality on tasks inside the model's capability frontier, but were less likely to reach correct answers on a task outside it [16]. For performance reporting, an error outside the frontier is not harmless: a fabricated number about a named employee can influence a decision. Huang et al.'s survey classifies such failures as factuality and faithfulness hallucinations and reviews retrieval and verification-based mitigations [23]. Retrieval-augmented generation conditions a model on retrieved documents to improve factuality and provide provenance [28], and Rashkin et al. define attribution to identified sources (AIS) as the property that every generated statement about the external world can be verified against a provided source, evaluated by trained human annotators [29]. WorkPlus applies a narrower, automatic form of this idea: the source is a numbered fact sheet, and only citations and numbers are verified.

Two strands of human–AI interaction research inform how such outputs are shown. Amershi et al. propose 18 guidelines, including making clear what the system can do and how well, supporting efficient correction and conveying the consequences of user actions [18]. Liao, Gruen and Miller, interviewing 20 practitioners, document the questions users ask of AI systems—why, why not, what data, how confident—and the gap between XAI algorithms and design practice [19]. For statistical anomaly detection, Blázquez-García et al. provide a taxonomy of time-series outlier methods, which highlights that point-outlier methods such as z-scores depend on assumptions about the series and its seasonality [17].

Comparative Analysis of Existing Studies

Table 2.1 compares the main strands of work against the dimensions relevant to this study. It is an analytical synthesis, not a feature benchmark.

Table 2: Comparative research matrix

Author and Year	Research Focus	Methodology	Main Findings	Limitations	Relevance
Madhumita et al. (2024) [5]	AI-powered performance management	Conceptual paper	AI can make appraisal continuous, predictive and personalised	No empirical evaluation reported; automation of judgement not problematised	Represents the "automate evaluation" approach this study avoids
Ravid et al. (2020) [9]	EPM review and typology	Integrative literature review	Monitoring characteristics (purpose, invasiveness, transparency) shape outcomes	Conceptual; pre-dates LLMs	Design variables for privacy-aware tracking
Ravid et al. (2023) [10]	EPM effects on work outcomes	Meta-analysis, k = 94, N = 23,461	No performance gain; higher stress; transparency helps attitudes	Mostly pre-2020 studies; heterogeneous designs	Justifies separating activity from performance
Forsgren et al. (2021) [13]	Developer productivity	Practitioner framework	Use ≥3 of 5 SPACE dimensions incl. perceptual	Not empirically validated as a scale	Basis for multi-indicator design and limits of activity data
Barredo Arrieta et al. (2020) [14]	Explainable AI	Survey and taxonomy	Transparent models support accountability	Generic; not HR-specific	Transparent weighted score and evidence display
Benlian et al. (2022) [20]	Algorithmic management	Expert panel discussion	Benefits and harms; mitigation via transparency and ethical design	Opinion-based	Frames the research gap

Three patterns emerge. The applied system paper reviewed here [5] emphasises automation without reported validation; behavioural research [9], [10] provides strong evidence about effects but no implementation guidance; and frameworks [13], [14] offer principles without an artefact that operationalises them. None of the works reviewed here evaluates an implemented tracker that grounds AI statements in records and separates presence from delivery.

Research Gap

Benlian et al. conclude that algorithmic management has both bright and dark sides and call for designs

that use transparency and ethical safeguards to mitigate harm [20]. The gap addressed here is narrower and testable: there is little published work that (a) implements activity measurement together with task contextualisation in a working system, (b) restricts the AI component to interpretation of server-computed facts with automatic removal of unsupported statements, and (c) enforces consent, scoping and retention in the server and evaluates these properties with reproducible tests. The study addresses this gap with one artefact; it does not claim that the artefact improves performance or well-being, which would require field evidence.

IV. RESEARCH METHODOLOGY

Research Approach

The study follows design science research (DSR), in which knowledge is produced by building and evaluating an artefact that addresses an identified problem [21]. The six activities of Peffers et al.'s design science research methodology [22] structure the work: problem identification (Sections 1.2 and 2), definition of objectives (1.3), design and development (3.3–3.6, 4), demonstration (the public deployment), evaluation (5) and communication (this paper). DSR was chosen over a purely empirical survey because the research questions concern whether specific design properties can be realised and verified; it was chosen over a controlled experiment with employees because ethical approval and an organisational partner were not available within the project.

Development was iterative. The repository history contains 42 commits (36 excluding merges) between 2 October and 11 November 2026, with feature work merged through pull requests and continuous integration running the unit tests on Windows, macOS and Linux; database-backed tests run in CI only on Linux and only for two test files (.github/workflows/ci.yml). The two authors are also the system's developers, so the evaluation reported here is a self-evaluation. Several design decisions recorded in the documentation arose from evaluation findings—for example, judging on-time delivery by submission rather than approval time, after review delays were found to make employees appear late (scenario S12).

System Requirements

Requirements were derived from the objectives and from the privacy principles in the literature [3], [9]. Table 3.1 lists the principal functional (FR) and non-functional (NFR) requirements together with the implementation that satisfies them; Section 5 reports how each was tested.

Table 3: Principal requirements

ID	Requirement	Realised by
FR1	Three roles with scoped access: admin (organisation), manager (own team), employee (self)	roleMiddleware.js; scope helpers (employeeScope, findEmployeeInScope)
FR2	Record sign-in and sign-out sessions	WorkSession model; POST /auth/login, /auth/logout; JWT sid claim
FR3	Task lifecycle with due dates, expected effort and manager review	Task model; taskController (todo → in-progress → review → done / changes)
FR4	Collect activity counts and application names with consent	Desktop agent trackers; POST /activity/snapshots with consent checks
FR5	Daily aggregation and dashboards	productivityService.recalculateDay → Performance; /performance/dashboard
FR6	AI-assisted reports and question answering with evidence	AI service /report; agentService pipeline; agentGrounding.js

ID	Requirement	Realised by
FR7	Employee control of tracking (pause, resume, stop)	Employee.tracking; GET/PUT /tracking; 15-s agent polling
NFR1	Data minimisation: no keystroke content, window titles or URLs	Activity and ApplicationUsage schemas; Zod schemas strip unknown fields
NFR2	Retention limits	TTL indexes (raw activity 90 d, camera labels); hourly screenshot purge (3 d)
NFR3	Security: hashed passwords, token revocation, rate limiting	bcrypt cost 12; tokenVersion; session check; limiters (10 failed logins per email and address/15 min; 30 AI requests/h)
NFR4	Graceful degradation when AI is unavailable	Report fallback (source: fallback); rule-based assistant
NFR5	Interactive latency for dashboards	Pre-aggregated Performance records; compound indexes

V. PROPOSED SYSTEM ARCHITECTURE

The architecture (Figure 3.1) separates collection, enforcement, computation and interpretation.

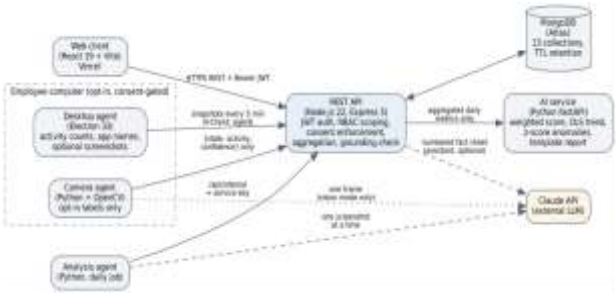


Figure 1: Overall architecture of WorkPlus as implemented. Solid arrows are always-present data paths; dashed and dotted arrows are optional, consent- or configuration-dependent LLM calls.

Following the EPM typology, which identifies purpose, invasiveness and transparency as levers [9], and the call for transparency safeguards in algorithmic management [20], the design places every policy decision in one component—the REST API—so that the desktop agent, web client and AI services cannot bypass it. Collection happens on the employee's computer and only for consented data types; enforcement, storage and aggregation happen in the API and the Python AI service; and interpretation by an LLM is optional and receives only derived facts.

Data Collection and Preprocessing

Data minimisation, which Ball recommends as a core policy principle [3], determined what the desktop agent collects. Every 60 seconds the agent builds a snapshot containing active and idle seconds (from Electron's powerMonitor idle time and lock events), mouse event counts (from cursor-position polling once per second through Electron's screen API) and keyboard event counts (from the uiohook-napi input hook, whose event payload is discarded) and seconds per foreground application name. Batches are uploaded every five minutes and queued locally while offline. Window titles, URLs and keystroke content are never captured. Screenshots are a separate opt-in, off by default, and require a second explicit agreement that the manager and administrators can view them.

Preprocessing occurs on the server before storage (Figure 3.2). Zod schemas reject malformed input (for example, active plus idle seconds exceeding the interval by more than a 5-second tolerance) and strip unknown fields (verified in Section 5.3); consent is checked per data type; records captured more than 90 seconds after tracking was paused or stopped are discarded; a unique index on (employee, capturedAt) makes uploads idempotent, and overlapping

intervals are discarded. Accepted snapshots trigger recalculation of one Performance document per employee per day, so that dashboards never scan raw data. For the assistant, a second preprocessing layer classifies task and session records into eleven data-quality codes (Section 3.6).

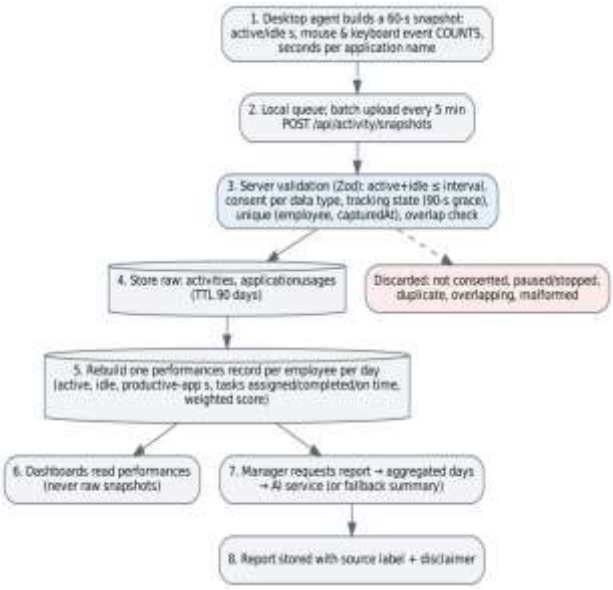


Figure 2: Data flow from desktop snapshot to dashboard and report, including the server-side validation that discards non-consented, paused, duplicate, overlapping or malformed records.

Performance Indicators

Following the SPACE argument that no single metric captures productivity [13], the study distinguishes five levels of evidence and assigns each indicator to one of them (Table 3.2). Two families of indicators exist in the code and are defined separately: those computed by the AI assistant over a period (agentMetrics.js), and the daily composite score stored in Performance records (performanceService.js). The distinction is enforced in the code: presence indicators are labelled as such in the assistant's fact sheet, and the assistant's prompt forbids calling anyone productive or unproductive on the basis of presence.

Table 4: Indicators, definitions and limitations

Level	Indicator	Definition (as implemented)	Principal limitation
Observable activity	Active ratio	activeSeconds ÷ (activeSeconds + idleSeconds) per day	Input activity is not work; can be gamed by mouse jigglers; reading/thinking counts as idle
Recorded system events	Sign-in sessions; recorded hours	Union of web sessions with a sign-out; missing sign-outs flagged, no duration counted	Presence only; overlapping browsers merged; closed browsers produce missing sign-outs
Observable activity	Productive-app ratio	Seconds in apps categorised "productive" ÷ active seconds	Category list is organisational judgement; same app can serve work or leisure
Task progress	Workload, pending, overdue, due soon	Tasks existing in period and not done before it; overdue = pending, past due, not submitted	Depends on managers entering due dates; task size varies
Task outcomes	Completion rate (assistant)	Tasks approved (completedAt) in period ÷ workload	Equal weight for unequal tasks; approval timing is the manager's

Level	Indicator	Definition (as implemented)	Principal limitation
Task outcomes	On-time rate (assistant)	Completed tasks submitted at or before due date ÷ completed tasks with a due date	Undefined without due dates; few tasks give unstable rates
Task outcomes	Daily task factors (composite)	Tasks due or completed that day (excluding tasks finished earlier): completed ÷ counted; on time = completed tasks with no due date or submitted by it ÷ completed	Tasks without a due date count as on time; on most days no task is due, so the factors are absent
Inferred performance	Weighted "activity score" (0–100), daily	0.30·active ratio + 0.30·productive-app ratio + 0.25·completion rate + 0.15·on-time rate; missing factors excluded and weights renormalised	Weights are design choices, not empirically derived; nominally 60% activity-based, but 100% activity-based on days without task factors (27 of 40 active days in the dataset)
Inferred performance	Trend; anomalies	OLS slope of daily scores (flat if |slope| ≤ 0.5 points/day); |z| ≥ 2.0 using population SD	Sensitive to non-working days and short windows (see 5.5)

The weighting formula is implemented identically in server/src/utils/productivityCalculator.js and ai-service/app/models/productivity_model.py, but the two aggregate a period differently: the dashboard and the fallback report average the daily scores of days with activity, whereas the AI service applies the formula once to the summed totals of all days in the period (a ratio of sums). Section 5.5 shows the consequence. Comparison between periods is reported only when both periods contain at least three tasks, which prevents a single completed task from being presented as a trend (scenario S9).

AI Agent Design and Workflow

The assistant implements a strict division of labour: the backend calculates, the model interprets, and the server checks every statement against evidence (Figure 3.3). This design follows from the hallucination and retrieval-augmentation literature, which identifies conditioning on authoritative content and post-hoc verification as mitigations [23], [28]. The statistical component uses established, inspectable techniques—least-squares trend slopes and z-score point-outlier detection—whose assumptions are documented in the time-series anomaly literature [17]; the detector returns no anomalies for fewer than three points or zero variance. With population standard deviation the largest attainable |z| in n points is √(n − 1), so the default threshold of 2.0 can first be reached with five points (the code's docstring states six, a documentation error).

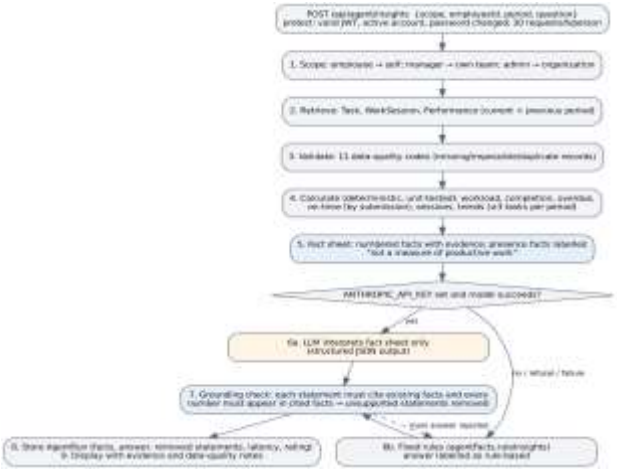


Figure 3: AI assistant workflow as implemented in server/src/services/agentService.js. The grounding check applies equally to LLM and rule-based answers.

The grounding check (agentGrounding.groundOutput) rejects a statement

when it cites no facts, cites a fact identifier not in the sheet, or contains a number that does not appear in the cited facts (including their rounded forms), the period length or—for the main answer only—the question. Record identifiers and exact timestamps are excluded as evidence for numbers, and digits in a task title count only when the title is quoted, so that "99 tasks are overdue" cannot pass because "99" occurs in an identifier. Calendar dates in the evidence are not excluded, however, so day, month and year components of a cited date can validate an unrelated number (Section 5.5). A rejected main answer is replaced by the rule-based answer, and removed statements are displayed and stored. A documented limitation is that numbers written as words are not checked. When ANTHROPIC_API_KEY is absent, or the model fails or refuses, fixed rules produce the answer and the interface states this.

Two further LLM components are optional and run outside the request path. The analysis agent classifies each consented screenshot into one of eleven generic categories (with "unclassified" set by the agent when the model declines) and a short activity label, using structured JSON output, and composes a daily summary from labels and aggregated metrics; the camera agent, in its default local mode, uses OpenCV face detection on the device and sends only a state label, and in vision mode sends a single frame to the model, which must return a generic state without identity, appearance or emotion.

Evaluation Methodology

Evaluation followed the DSR guideline that the utility and quality of an artefact must be demonstrated through well-executed evaluation methods [21]. The evaluators are the system's developers. Seven evaluation activities were performed in the authors' test environment on 22 November 2026: (E1) execution of all automated test suites in the repository; (E2) execution of the project's AI assistant evaluation harness (13 hand-computed scenarios and 10 API/authorisation cases); (E3) 23 black-box functional and negative API tests written for this study; (E4) data-ingestion robustness tests (duplicate, overlapping, paused and inconsistent snapshots); (E5) latency (100 sequential requests per read endpoint, and 50 or 30 for the assistant and report endpoints, after 5 warm-up calls) and throughput (10 concurrent clients for 10 seconds) measurements; (E6) inspection of eight generated reports (four employees × weekly and monthly periods) for factual and temporal consistency, with an ablation that excludes zero-activity days; and (E7) an adversarial perturbation test of the grounding check, in which valid rule-based statements from all 13 scenarios were systematically corrupted (numbers altered, citations removed, invalid or unrelated facts cited, numbers written as words, unsupported qualitative claims added) and the detection and false-rejection rates were recorded. In addition, the public deployment was inspected read-only. Latency is reported as mean, standard deviation, median (p50) and 95th percentile (p95), with throughput = completed requests ÷ duration and error rate = responses with status ≥ 400 ÷ completed requests. Classification metrics (accuracy, precision, recall, F1) were not used for the AI components: the system contains no trained classifier with ground-truth labels, and the LLM components could not be scored without a live model and human-labelled data. Detection rates in E7 are reported only for the deterministic grounding check, against perturbations whose ground truth is known by construction.

Ethical Considerations

Because monitoring is associated with stress regardless of configuration [10], the study treated the system itself as ethically sensitive. No real employee data was collected: all evaluation data is synthetic and labelled as such. Credentials used in tests are the publicly documented demo accounts of a local, throwaway database; no accounts were created on the public deployment. The design embeds developmental rather than punitive framing: AI recommendations are phrased as supportive check-ins, reports carry an "AI-generated … review before acting" disclaimer, and the assistant produces no performance score. These choices are consistent with the NIST AI Risk Management Framework's emphasis on human oversight and context-specific risk management [24]. Because the authors evaluated their own system, results should be read as developer verification rather than independent

validation. Any future field study would require informed consent, institutional ethics review and compliance with India's Digital Personal Data Protection Act, 2023 [26].

VI. SYSTEM DESIGN AND IMPLEMENTATION

Technology Stack

WorkPlus is a monorepo using npm workspaces for the three JavaScript applications and separate Python packages for the three Python services (Table 4.1). Versions are those declared in the manifests at commit d3ee6ef.

Table 5: Technology stack

Component	Technologies (declared versions)	Responsibility
client/	React 19, React Router 7, Vite 6, lucide-react; deployed on Vercel	Role-specific dashboards for admin, manager and employee
server/	Node.js ≥ 22, Express 5, Mongoose 8, Zod 3, jsonwebtoken 9, bcryptjs 2, @anthropic-ai/sdk; deployed on Render	Authentication, authorisation, consent enforcement, storage, aggregation, AI assistant and grounding
Database	MongoDB ≥ 7 (Atlas in production)	13 collections with compound and TTL indexes
desktop-agent/	Electron 33, uiohook-napi, get-windows, electron-builder 25	Activity snapshots, consent wizard, tracking control, optional screenshots
ai-service/	Python ≥ 3.11, FastAPI, Pydantic 2 (standard-library statistics)	Weighted score, OLS trend, z-score anomalies, template-based report text
analysis-agent/	Python, Anthropic SDK (default model id claude-opus-5-5)	Daily screenshot classification and whole-day summary (optional)
camera-agent/	Python, OpenCV 4 (headless), Anthropic SDK	Opt-in camera state labels; local mode by default (optional)
CI/CD	GitHub Actions (Windows, macOS, Linux); Render blueprint; Vercel	Tests on every push; agent installers per release tag

System Architecture and Modules

The API follows a routes–controllers–services–models layering. Fifteen route modules expose 68 route handlers plus a health endpoint, mounted under /api/auth, /employees, /users, /tasks, /activity, /performance, /reports, /consent, /tracking, /screenshots, /analysis, /camera, /agent, /downloads and /internal. Pure computation is isolated in utility modules (productivityCalculator, agentMetrics, agentFacts, agentGrounding, workStatus, taskMonitor), which makes it unit-testable without a database. The /internal routes, used only by the analysis agent, are protected by a shared service key compared in constant time (crypto.timingSafeEqual). The AI service exposes /productivity, /trends, /anomalies and /report and receives only aggregated daily metrics.

Each registration creates a separate organization, and all queries are scoped to it.

Table 6: Implementation inventory (verified against source code and tests)

Class	Features
A. Implemented and verified by tests in this study	JWT authentication with session-linked revocation; role and team scoping; task lifecycle with review; work sessions; activity ingestion with consent, tracking-state, duplicate and overlap handling; daily aggregation and dashboards; AI-service statistical reports and fallback; AI assistant in rule-based mode with grounding check; rate limiting; multi-organisation isolation
B. Implemented, with configuration or operational limitations	AI assistant LLM mode (requires ANTHROPIC_API_KEY; not exercised here); analysis agent and camera vision mode (require API key and consent; unit-tested with mocks only); desktop agent (unit-tested modules; GUI and OS hooks not executed in this study); screenshot storage (ephemeral disk on Render free plan unless Cloudinary/disk configured); in-memory rate limiter (per process); hosted API cold starts
C. Partially implemented	Human usefulness rating (stored on AgentRun, no ratings collected); LLM evaluation harness Part C (exists, not run); Postman collection of 338 requests (present, not executed in this study)
D. Proposed or planned	LLM-written narrative in the AI-service report (code comment marks an "LLM HOOK"); field study with consenting employees; user studies of comprehension
E. Not implemented (some by design)	Keystroke content, window titles and URLs (excluded by design); trained machine-learning models; email-based password reset; formal compliance certification or data-protection impact assessment

Database Design

The schema (Figure 4.1) contains thirteen Mongoose models. User holds identity and role; Employee holds the employment relation, manager link, consent record, camera consent and tracking state. Raw signals (Activity, ApplicationUsage, CameraObservation, Screenshot) are separated from derived records (Performance, DailyAnalysis, Report) so that retention can differ: raw activity and application usage carry a 90-day TTL index on createdAt, camera labels expire after CAMERA_RETENTION_DAYS (30 days by default), and screenshots are removed three days after capture by an hourly job, while daily aggregates are kept. Unique compound indexes on (employee, capturedAt) and (employee, date) enforce idempotent ingestion and one aggregate per day. AgentRun stores every assistant question, fact sheet, answer, removed statements, latency and optional rating, providing an audit trail.

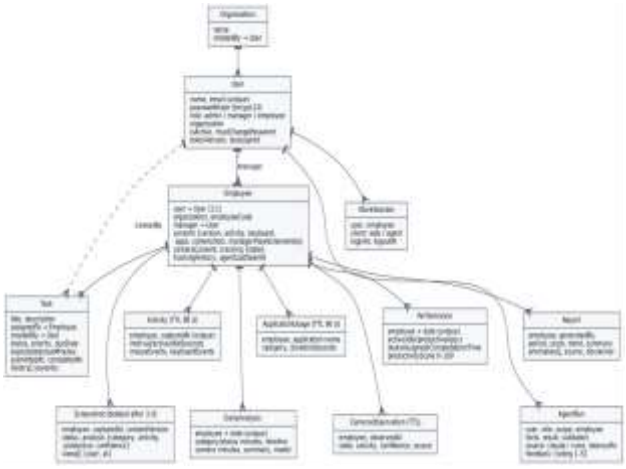


Figure 4: Entity–relationship view of the 13 MongoDB collections (key fields only), derived from server/src/models.

Employee Login and Logout

POST /api/auth/login verifies the bcrypt hash, opens a WorkSession and issues a JWT containing the user id, role, token version and session id (sid). Web tokens expire after three days by default; tokens issued to employees signing in through the desktop agent expire after thirty. POST /api/auth/logout records the sign-out time; because every protected request checks that the session has not been closed, a token cannot be reused after sign-out even before it expires (verified in test F06). Accounts created by an administrator start with a temporary password, and until it is changed the server permits only /auth/me, /auth/password and /auth/logout. Ten failed sign-ins for the same email address from the same network address within 15 minutes trigger a temporary block.

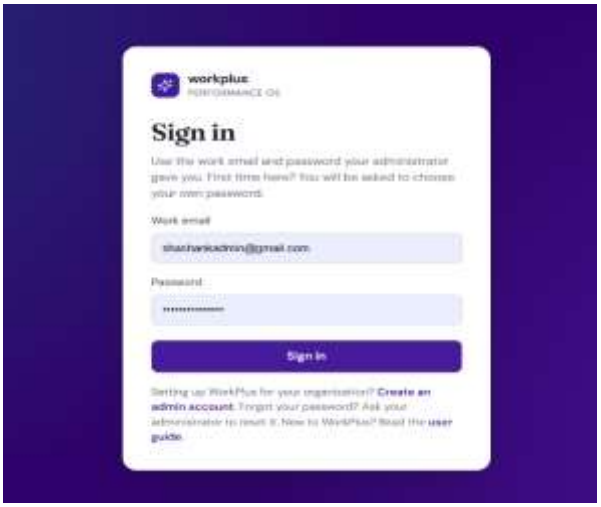


Figure 5: Application login interface (/login).

König notes that monitoring outcomes depend on how monitoring is launched and continued, including whether workers can see and influence it [11]. In WorkPlus, signing in on the website sets the tracking state to active, Pause/Resume is available both in the dashboard top bar and in the agent, and signing out of the dashboard stops tracking. The agent polls GET /api/tracking every 15 seconds and follows the server state; the server discards activity captured more than 90 seconds after a pause or stop (60 seconds for camera labels) and refuses screenshots captured afterwards with HTTP 409. Consent is required independently: an active state never starts a tracker the employee has not agreed to (Figure 4.3).

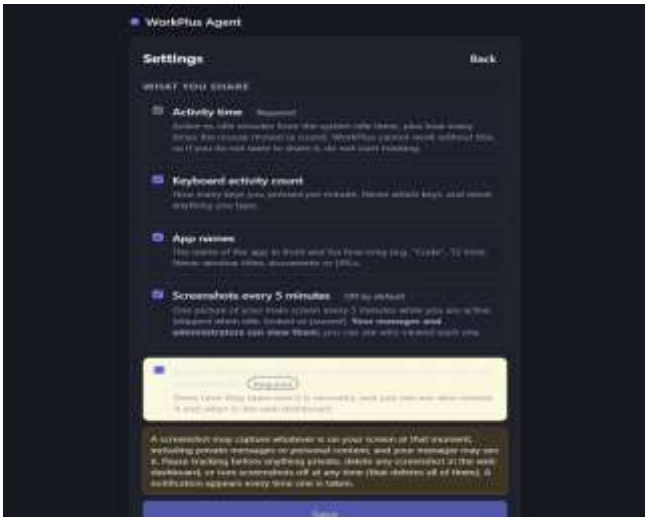


Figure 6: Desktop agent consent and data-sharing choices (Desktop agent).



Figure 7: Employee dashboard with tracking control (/employee).

Task and Deadline Management

Parent-Rocheleau and Parker show that algorithmic goal setting and performance management alter job demands, moderated by human influence over the algorithm [15]; Benlian et al. similarly recommend keeping humans accountable for consequential decisions [20]. The task lifecycle (Figure 4.5) implements this as a human-in-the-loop state machine. Managers create tasks with title, description, priority, due date and expected minutes for members of their own team. Assignees move tasks to in-progress, log actual minutes and submit for review with a note; the server records submittedAt. Only the manager can approve (setting

completedAt) or request changes with feedback, and an employee's attempt to set status done is rejected (test F16). Every step is appended to the task's history. On-time delivery is judged by submission time, so that a slow review never makes an employee late.



Figure 8: Task lifecycle as enforced by server/src/controllers/taskController.js.

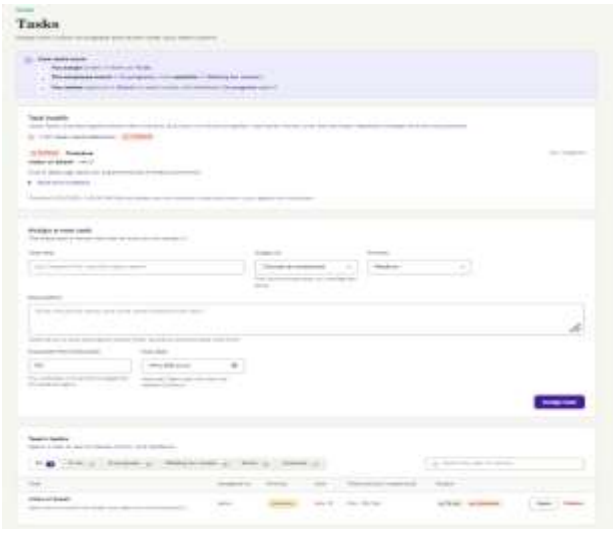


Figure 9: Manager task management interface (/manager/tasks).



Figure 10: Employee task view and submission for review (/employee/tasks).

Manager Dashboard

The manager dashboard (Figure 4.8) reads pre-aggregated Performance records and shows tracked time, task counts and a per-employee score whose interface label is "Activity score". The terminology is significant: the client labels the composite as activity, but the underlying field is named productivityScore and the AI-service report text describes it as a "productivity score" (Section 6.1). A task monitor surfaces overdue, due-soon and review-overdue tasks, distinguishing work waiting for the employee from work waiting for the manager. Managers see only employees whose manager field is their own user id; administrators see their organisation. The dashboard does not show how each score was computed; Liao et al. found that users of AI systems ask what data an output is based on and why it takes its value [19], a need that the assistant's evidence display (Section 4.7) addresses but the dashboard does not.

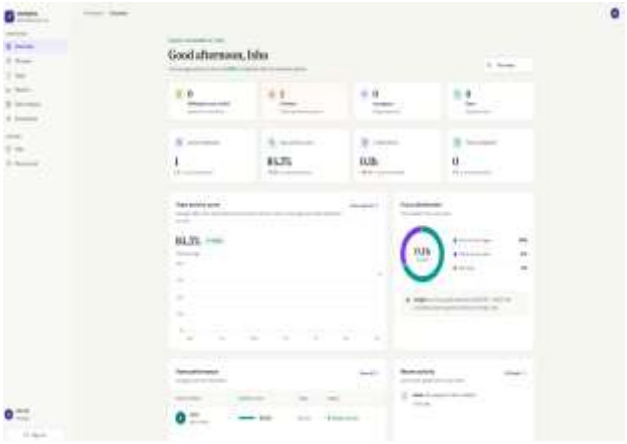


Figure 11: Manager dashboard (/manager).

AI Agent Integration

The assistant page is available to every role, scoped as in Section 3.6. Amershi et al.'s guidelines call for making clear how well the system can do what it does and for graceful handling of failure [18]. The implementation responds to both: every answer states whether it came from the LLM or from fixed rules, shows each observation with the facts that support it, lists data-quality issues and shows statements removed by the evidence check; if the model is unavailable, times out or refuses, the rule-based answer is returned. Each person may make 30 requests per hour. The model receives names, employee codes, task titles, dates and counts, but never task descriptions, comments, screenshots or passwords, and the API key is held only in the server environment.

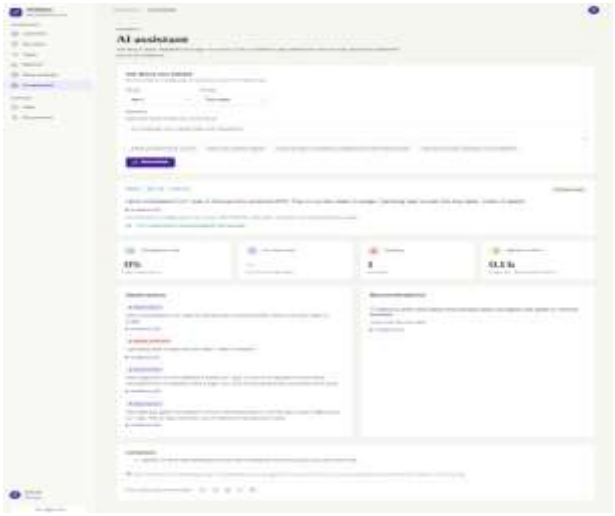


Figure 12: AI assistant answer with evidence (/manager/assistant).

Reports are generated separately. When a manager requests a weekly or monthly report, the server sends that employee's daily Performance records to the AI service, which computes the score, OLS trend, anomalies, highlights and recommendations and returns template-generated text; no LLM is involved in this path. If the AI service is unreachable, the server writes a basic summary marked source: fallback. Both are stored with a disclaimer (Figure 4.10).

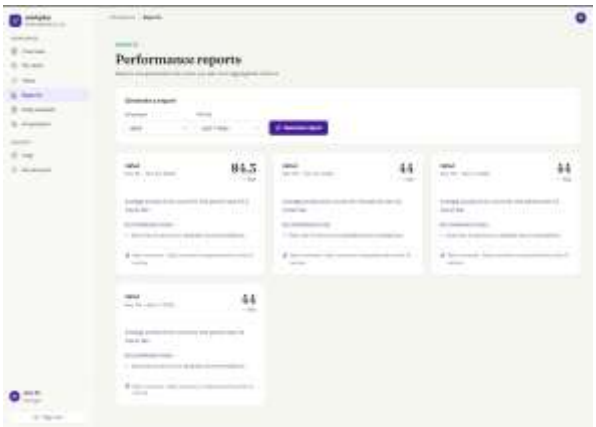


Figure 13: AI-generated performance report (/manager/reports).

Security and Access Control

Monitoring policy guidance stresses that access to monitoring data should be limited to what is necessary [3], and the EPM literature treats who receives monitoring data as a characteristic that shapes its effects [9]. Access control in WorkPlus therefore has two aims: preventing unauthorised access and preventing data from being enumerated across scopes. In the other direction, Jarrahi et al. warn that algorithmic management creates information asymmetries that disadvantage workers [8]; the employee-facing screenshot view log and employees' access to the assistant are partial responses. The request pipeline (Figure 4.11) verifies the JWT signature and expiry, re-reads the user so that deactivation and role changes apply immediately, compares the token version (incremented on password change, reset or deactivation), checks that the session is open, enforces the temporary-password restriction, and applies role and scope rules in roleMiddleware.js. Requests for resources outside the caller's scope return 404 rather than 403, so that the existence of other employees is not disclosed; an employee who passes another employee's id silently receives only their own data (test F08).

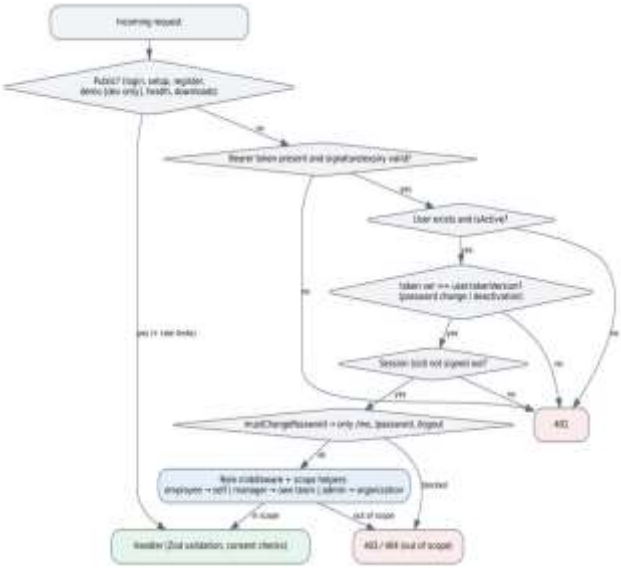


Figure 14: Authentication and authorisation workflow implemented in server/src/middleware/authMiddleware.js and roleMiddleware.js.

Other controls include bcrypt password hashing with cost factor 12, a 256 KB JSON body limit, a CORS allow-list restricted to the configured client origin, input validation on every write endpoint, and disabling of password-free demo sign-in whenever NODE_ENV is production (confirmed on the live

deployment, where /api/auth/demo reports enabled: false). For screenshots, each view by a manager or administrator appends {user, at} to the record, the employee sees who viewed each image and can delete it, and screenshots taken under consent version 1 ("managers never see screenshots") are never returned to managers.



Figure 15: Screenshot view log visible to the employee (/employee/screenshots).

VII. EXPERIMENTAL EVALUATION AND RESULTS

Experimental Setup

All local measurements were taken on 22 November 2026 by the authors in the environment described in Table 5.1; the raw outputs (environment capture, test logs and JSON results) accompany the experimental test report. The API, AI service and database ran on the same machine, so network latency is excluded and the results represent server-side processing on modest hardware rather than production latency. MongoDB 8.3.7 was used because it was the version obtainable in the environment; the project requires version 7 or later.

Table 7: Test environment

Item	Value
Hardware	Intel Xeon processor @ 2.10 GHz, 2 vCPUs, 8 GB RAM (8,024 MB; cloud Linux container)
Operating system	Ubuntu 24.04.5 LTS
Runtimes	Node.js 22.23.5, npm 10.9.9, Python 3.13.18
Database	MongoDB 8.3.7 (local, single node, default configuration)
Code under test	Repository commit d3ee6ef (main branch, 11 Nov 2026)
Server configuration	NODE_ENV=development, APP_TIMEZONE=Asia/Kolkata, AI_SERVICE_URL=local FastAPI (uvicorn), ANTHROPIC_API_KEY unset (rule-based assistant)
Benchmark client	Node.js fetch on the same host (bench.mjs, supplied with the test report; not part of the repository)
Public deployment (read-only checks)	Vercel client and Render API, accessed from the authors' computer through a browser; observations recorded in a log

Dataset Description

The dataset is synthetic and was produced by the project's seed script (server/src/scripts/seed.js), which uses a linear congruential generator with a fixed seed (42). It models one administrator, one manager and four employees (EMP-001 to EMP-004) with different "focus" parameters (0.70–0.92) over the 14 days ending on the run date, with activity only on working days and during working hours.

Because the window is anchored to the run date, the values are reproducible only for a given run date: which weekdays fall inside a weekly window—and therefore the weekend effects reported in Section 5.5—change with the day the seed is run. All runs reported here used 22 November 2026 (a Sunday).

Table 5.2 lists the collection sizes counted immediately after seeding. Eight of the 40 sign-in sessions deliberately lack a sign-out, simulating closed browsers. The dataset contains no real personal data, but it encodes the developers' assumptions about work patterns and is far smaller than an organisational dataset.

Table 8: Synthetic evaluation dataset (counts immediately after seeding)

Collection	Records	Notes
users / employees	6 / 4	Admin, manager, four employees (EMP-001 to EMP-004)
activities	320	Activity snapshots on working days
applicationusages	640	Application names from a fixed list (e.g., Code, Figma, Slack, Chrome)
performances	56	One per employee per day (4 × 14); scores 0–100, mean 69.3
tasks	32	done 16, review 6, in-progress 4, todo 6
worksessions	40	8 without sign-out (missing-logout case)
cameraobservations / dailyanalyses	960 / 10	Sample labels only; no images; labelled as sample data
screenshots, reports, agentruns	0	Created during testing only

The evaluation harness (E2) uses a second, independent dataset: thirteen hand-constructed scenarios in server/evaluation/agentScenarios.js whose expected values were worked out by hand, covering completion arithmetic, overdue and pending work, missing deadlines, missing sign-ins and sign-outs, impossible and duplicate records, heavy and light workloads, absence of data, presence versus productivity, and on-time judgement at submission.

Functional Testing

Table 5.3 summarises execution of the repository's automated suites (E1). All 190 tests passed; because the Node.js runner also counts subtests, the server suite reports 150 test points for its 98 top-level tests, giving 242 test points in total. Twenty-three of the server's top-level tests (75 test points), including the end-to-end admin → manager → employee workflow, screenshot scoping and task-monitor API tests, require MongoDB; they were run against a throwaway local instance and are skipped without one.

Table 9: Automated test suites (E1)

Suite	Runner	Tests (test points)	Passed	Failed	Duration
server (23 top-level tests need MongoDB)	node --test	98 (150)	98 (150)	0	45.4 s
ai-service	pytest	34	34	0	0.44 s
analysis-agent (LLM mocked)	pytest	25	25	0	1.90 s
camera-agent (camera/LLM mocked)	pytest	20	20	0	1.11 s
desktop-agent (pure modules)	node --test	13	13	0	0.23 s
Total		190 (242)	190 (242)	0	

The evaluation harness (E2) reported 88 of 88 checks passing in Part A (calculations and data quality across scenarios S1–S13) and 10 of 10 cases passing in Part B (sessions, authentication, employee and manager scoping, input validation, rule-based fallback, rating privacy and rate limiting). Part C, which measures an LLM's groundedness, coverage, wording and resistance to an instruction embedded in a task title, was not run.

Table 5.4 lists the black-box checks written for this study (E3), which use only documented endpoints. All 23 passed in the reported run. During development of the test script, several checks failed because the script read the wrong response fields; the console output of those runs was not retained. The script was corrected and the database reseeded before the reported run.

Table 10: Black-box functional and negative API tests (E3)

ID	Area	Test	Expected	Result
F01–F03	Authentication	Wrong password; no token; tampered JWT	401 each	PASS
F04	Authentication	Valid token returns own profile and role	manager	PASS
F05–F06	Login/logout	Sign-out closes session; token unusable afterwards	204; then 401	PASS
F07	Role-based access	Employee lists managers (admin-only)	403	PASS
F08	Role-based access	Employee passes another employee's id	Own data only	PASS
F09	Role-based access	Employee requests team-scope AI insights	403	PASS
F10	Role-based access	Unknown/out-of-scope employee id	404	PASS
F11	Service API	/internal without service key	401	PASS
F12–F14	Task management	Manager creates task; invalid title; employee creates task	201; 400; 403	PASS
F15–F18	Task lifecycle	Start; self-approve attempt; submit; manager approves	in-progress; 400; review; done	PASS
F19–F20	AI assistant	No API key → rules, labelled; presence facts carry label	rules; label present	PASS
F21	Reporting	Report via AI service with source and disclaimer	ai-service + disclaimer	PASS
F22–F23	Activity ingestion	Non-employee upload; empty batch	403; 400	PASS

Robustness to faulty input (E4) was tested with a desktop-agent sign-in whose tracking state was explicitly set to active. A valid snapshot was stored; an identical second upload was accepted but stored nothing (idempotent); a snapshot starting inside the recorded interval was discarded ("It overlaps time already recorded"); a snapshot carrying extra fields (keys, windowTitle, url) was stored, and inspection of the stored document confirmed that only the schema fields were persisted; a snapshot whose

active seconds exceeded its interval was rejected with HTTP 400 ("active + idle exceeds the interval"); and after tracking was paused, a new snapshot was discarded ("Tracking was paused or stopped at that time"). An earlier attempt at these tests used a web sign-in without setting the tracking state, so every upload was discarded as paused; that attempt is reported in the test report and not counted. With the AI service stopped, all 30 report requests succeeded with source: fallback.

Performance Metric Evaluation

Correctness of the indicators was verified by the harness scenarios, whose expected values were computed by hand: for example, in scenario S8 a week with 16 workload tasks, 13 completed and 12 on time yields a completion rate of 81.3% and an on-time rate of 92.3%, and the comparison with the previous week (50.0%) is reported as +31.3 points because both periods contain at least three tasks. The harness covers the assistant's indicators; the daily composite score is covered by unit tests of productivityCalculator. All implemented indicators fall within the activity and performance dimensions of the SPACE framework [13]; satisfaction, collaboration and flow are not measured. In the freshly seeded data, 27 of the 40 employee-days with activity (67.5%) had no task due or completed, so on those days the composite score was computed from activity factors alone.

System performance (E5) is reported in Table 5.5 and Figure 5.1. Median latency of the authenticated read endpoints ranged from 5.0 ms to 17.3 ms, and 95th-percentile latency was at or below 27.0 ms; the task monitor showed one outlier of 217.9 ms in 100 requests. The manager dashboard was the slowest read endpoint because it aggregates team-wide records. Report generation through the AI service had a median of 11.5 ms, and the fallback path, which first fails to reach the AI service, had a median of 13.5 ms. A preliminary run in the same environment gave medians 0.2–4.8 ms higher, indicating run-to-run variation of that order. Under 10 concurrent clients, the manager dashboard sustained 95.6 requests per second and the task list 199.1 requests per second, with no errors (Table 5.6).

Table 11: Endpoint latency (ms), local environment

Endpoint	n	Mean	SD	p50	p95	Max
GET /health (no auth)	100	0.9	0.3	0.8	1.6	2.4
GET /performance/dashboard (manager)	100	18.5	4.5	17.3	27.0	34.9
GET /performance/dashboard (employee)	100	13.7	2.2	13.1	18.1	23.5
GET /tasks (manager)	100	9.3	2.3	8.8	12.8	21.6
GET /tasks/monitor (manager)	100	9.4	21.4	6.5	10.0	217.9
GET /employees (manager)	100	5.4	1.2	5.0	7.7	10.6
GET /reports/overview (manager, one employee)	100	8.1	1.3	7.8	9.8	17.4
GET /agent/metrics (team, week)	100	8.0	1.0	7.9	9.8	13.3
POST /agent/insights (rule-based)	50	10.7	1.8	9.8	15.1	16.7
POST /reports (AI service up)	30	12.5	4.8	11.5	15.0	37.5
POST /reports (AI service down → fallback)	30	15.3	–	13.5	29.0	33.9

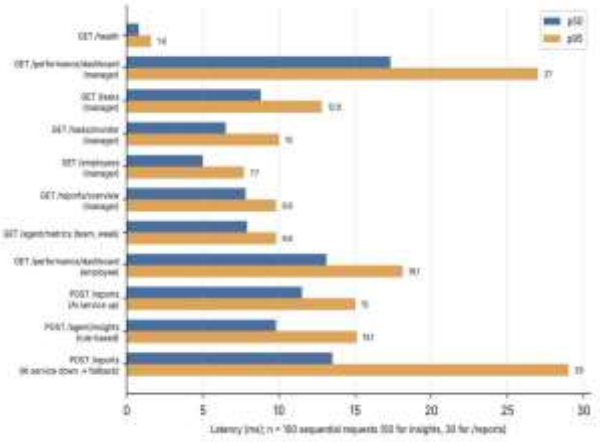


Figure 16: Median and 95th-percentile latency per endpoint (local, single host, synthetic dataset).

Table 12: Throughput under 10 concurrent clients for 10 s

Endpoint	Requests	Throughput (req/s)	Error rate	p50 (ms)	p95 (ms)
GET /performance/dashboard (manager)	956	95.6	0%	104.0	154.1
GET /tasks (manager)	1,991	199.1	0%	49.1	67.9

The public deployment was checked read-only from a browser, and the observations were recorded in a log. The health endpoint reported the database as connected. After an idle period, the first two requests took 19.6 s in total, consistent with the free hosting tier's documented sleep behaviour; ten subsequent health requests took 274–818 ms (median 318 ms), which includes internet round-trip time. Unauthenticated access to /api/tasks returned 401, password-free demo sign-in was disabled, and the download endpoint listed desktop-agent installers for version 0.1.4. Authenticated pages of the live site were not inspected because no test account existed there and creating accounts on a public system was outside the study's ethical scope.

AI Output Evaluation

Because no LLM key was used, AI output evaluation covers the grounding check, the rule-based assistant and the statistical report generator. Rule-based answers are composed from the same fact sheet that the check uses, so their passing the check (0 statements removed in 13 scenarios and 50 benchmark requests) is expected by construction and is not evidence that the check would catch an LLM's errors. The perturbation test (E7) addresses this by measuring what the check detects (Table 5.7).

Table 13: Grounding-check perturbation test (E7), statements from scenarios S1–S13

Case	n	Rejected	Rate	Interpretation
C0 Valid statement, unchanged (control)	54	0	0%	No false rejection
C1 Valid number written with one decimal (control)	50	0	0%	Rounded forms accepted
P1 One number altered (+7)	50	46	92%	Misses when the new value matches a date component or another value in the cited facts
P1b One number scaled (×10 + 3)	50	46	92%	As P1
P2 All citations removed	54	54	100%	Detected

Case	n	Rejected	Rate	Interpretation
P3 Non-existent fact id added	54	54	100%	Detected
P5 Numbers kept, citation swapped to unrelated fact	50	50	100%	Detected
P4 Wrong number written as a word	44	0	0%	Not detected (documented limitation)
P6 Unsupported qualitative judgement, valid citation	54	0	0%	Not detected: the check verifies numbers and citations, not meaning

The missed number alterations had two causes. In most, the substituted value matched the day or month component of a date in the cited evidence—for example, "completed 11 of 4 tasks" passed because a cited task was dated 2026-10-11—so any number from 1 to 31 can pass whenever a dated record is cited. In the others, the substituted value equalled another value in the cited facts, such as a session count changed to the recorded number of hours. The check therefore enforces a necessary but not sufficient condition: in the terminology of attribution evaluation, it verifies that numbers are attributable to some cited source but not that the statement as a whole is supported [29], which requires human or model-based judgement.

Inspection of generated reports (E6) covered all four employees for weekly and monthly periods, generated on a freshly seeded database before any other test wrote to it (Table 5.8). Two defects were found. **Defect 1 (temporal):** the AI service receives every daily record in the period, including weekend days with no activity. All eight reports described the trend as "down", and all four anomalies flagged fell on days with no recorded activity. Because the evaluation date was a Sunday, every period ended with two inactive days, which drives the least-squares slope downwards. For EMP-001, the weekly report stated "Performance is declining (−17.0 points per day)" and recommended a "supportive check-in" because of the declining trend, although the employee's five working days had a flat trend (see the ablation below); for EMP-004, both Sundays in the monthly window were flagged as anomalies ($z = -2.35$). Conversely, an inactive day can score 100 if a task was completed on time that day: the reports for EMP-002 and EMP-004 named Saturday 21 November, a day with no recorded activity, as the "best day". This is the seasonality problem that point-outlier methods are known to face [17]. **Defect 2 (aggregation):** the AI service applies the weighting formula to the period's summed totals, whereas the dashboard and fallback average daily scores of days with activity. The resulting scores differ by 2.2–23.8 points (EMP-001 weekly: 84.8 versus 95.8). The AI-service value 84.8 equals $0.30 \times 0.92 + 0.30 \times 0.99 + 0.25 \times 1/2 + 0.15 \times 1.00$, that is, it includes the period's task completion rate (1 of 2), while most daily scores average only activity factors because no task was due on those days.

Table 14: Generated reports (AI service) compared with the dashboard aggregation, clean database

Employee	Period	Days (no activity)	AI-service score	Mean of daily scores	Trend	Anomalies (on no-activity days)
EMP-001	week	7 (2)	84.8	95.8	down	0 (0)
EMP-001	month	14 (4)	91.5	94.9	down	0 (0)
EMP-002	week	7 (2)	80.7	83.0	down	1 (1)
EMP-002	month	14 (4)	81.4	83.6	down	0 (0)

Employee	Period	Days (no activity)	AI-service score	Mean of daily scores	Trend	Anomalies (on no-activity days)
EMP-003	week	7 (2)	57.3	81.1	down	0 (0)
EMP-003	month	14 (4)	73.0	78.5	down	0 (0)
EMP-004	week	7 (2)	68.0	80.0	down	1 (1)
EMP-004	month	14 (4)	72.5	82.8	down	2 (2)

An ablation re-sent the same records to the AI service with zero-activity days removed. With all days included, the service reproduced every stored report score exactly. With inactive days excluded, the trend changed from "down" to "flat" or "up" in seven of the eight reports and remained "down" only for EMP-004's monthly report. The total number of anomalies stayed at four, but they moved from inactive days to working days, because removing the inactive days narrows the distribution and exposes other outlying working days. Excluding inactive days therefore removes the weekend artefact but is not by itself a complete correction; a working-day calendar and a minimum-window rule are also needed. Scores changed by −8.2 to +23.8 points, because some inactive days still carry task counts (tasks due at the weekend). An earlier run of this inspection was discarded after the second review round found that it had been executed after the black-box and ingestion tests had added a task and activity records to the database; the values above come from the clean re-run. Two further observations concern presentation: the report text calls the composite a "productivity score" although the interface labels it "Activity score", and the template wording ("strong productivity score") adds an evaluative adjective to a number whose activity share is high.

Comparative Analysis

A quantitative comparison with published AI appraisal systems is not possible: the applied work reviewed [5] reports no evaluation on a shared dataset. Table 5.9 therefore compares design properties that can be stated from the source and the code. The comparison shows a difference in where AI is placed; it does not show that WorkPlus produces better appraisals.

Table 15: Design-level comparison

Property	AI appraisal approach [5]	WorkPlus (this study)
Role of AI	Continuous, predictive appraisal and personalised feedback	Interpretation of server-computed facts; no AI-generated score in the assistant (the dashboard composite is a fixed formula)
Evidence for statements	Not reported	Numbered facts; numeric grounding check; removed statements shown
Activity vs performance	Continuous monitoring as input to appraisal	Separate constructs in the assistant; presence explicitly labelled; composite still mixes them
Privacy controls	Not detailed	Per-type consent, pause, scoping, retention for raw data, view log (server-enforced)
Behaviour without AI	Not reported	Rule-based answers and fallback reports, labelled
Reported evaluation	Conceptual; none reported	Automated tests, harness, black-box tests, perturbation test, latency, defect analysis (developer-run)

VIII. RESULTS AND OBSERVATIONS

- **Functional correctness.** All 190 automated tests (242 test points) and all 23 black-box API tests passed; the task lifecycle, session revocation and role scoping behaved as specified.
- **Metric correctness.** All 88 hand-computed checks matched, including the edge cases of missing timestamps, duplicates and empty data.
- **Robustness.** Duplicate, overlapping, extra-field, paused and inconsistent snapshots were handled without corrupting aggregates; AI unavailability produced labelled fallbacks.
- **Grounding.** The check detected citation errors fully and number alterations in 92% of cases with no false rejections, but cannot detect numbers written as words, numbers matching date components, or unsupported qualitative claims.
- **Performance.** Median latency of 5.0–17.3 ms for reads and 11.5 ms for report generation on two vCPUs; 96–199 req/s at concurrency 10 without errors.
- **Defects found.** Inactive days distort AI-service trends and anomalies (8 of 8 reports trending down; all 4 anomalies on inactive days); the AI service and dashboard aggregate scores differently (gaps of 2.2–23.8 points).
- **Missing evidence.** No LLM output evaluation, no usability or comprehension study, no real-workforce data, no independent evaluator.

IX. DISCUSSION

Interpretation of Results

RQ1 is answered positively for the AI assistant and only partly for the system as a whole. The assistant keeps activity, task events and derived indicators separate and labels presence explicitly. The daily composite score, however, combines activity and task factors and, on the two-thirds of active days with no task due or completed, reduces to an activity-only measure. It is labelled "Activity score" in the interface, stored as productivityScore and described as a "productivity score" in report text. The assistant refuses to score people while the dashboard and reports do score them. This internal tension is the most important design issue for the next iteration: either the composite should be renamed consistently and confined to activity factors, or task outcomes should be reported only as separate rates. **RQ2** is answered for the deterministic check, not for an LLM. The perturbation test shows that the check reliably detects missing, invalid and mismatched citations and most altered numbers, and does not falsely reject valid statements, but it cannot detect numbers written as words, numbers that coincide with date components or other cited values, or unsupported qualitative judgements—the last being the main risk in performance reporting. Whether the full pipeline keeps an LLM's statements grounded remains untested until harness Part C is run with a live model and human attribution judgements of the kind proposed by Rashkin et al. [29]. Failure of the AI components degraded gracefully in all tests. The report defects illustrate the limit of numeric grounding: a statement can be numerically correct with respect to its inputs and still misleading.

RQ3 is answered for the properties that can be tested in software: consent per data type, tracking state, scope isolation, token revocation and input validation were enforced at the server and verified by tests. Section 7 identifies properties that the software does not enforce, including retention of derived records and the voluntariness of consent. **RQ4** is answered positively for functionality, indicator correctness and latency under the stated conditions, with two report defects and several unmeasured qualities.

Comparison with Existing Studies

The applied literature presents AI as a means of automating appraisal [5]. The present study takes a different position, consistent with evidence that monitoring does not improve performance [10] and with guidance that humans should retain decision rights [15], [20]: in the assistant, AI is used to explain records to a human decision-maker, and the system refuses to infer effort or productivity from presence. The behavioural literature suggests that transparency of this kind should improve employee attitudes towards monitoring [10], but this study provides no evidence on that point.

Practical Implications

For organisations, the results suggest three practices. First, task data entered by managers—due dates, expected effort and timely reviews—carries more information about delivery than any activity signal, and the system's indicators are only as good as that data; the assistant's data-quality warnings make this visible. Second, AI-generated recommendations should be treated as prompts for conversation, as the report disclaimer states, not as conclusions; the weekend defect shows how a fluent recommendation can rest on an artefact. Third, deployment decisions such as enabling screenshots or camera checks change the invasiveness of monitoring, which the literature identifies as a key moderator [10]; the software makes these opt-in, but the organisation decides whether to ask for them. For developers, the pattern of server-side computation, constrained interpretation and automatic verification is transferable to other domains where LLMs summarise records about people, provided that its limits (Table 5.7) are understood.

Limitations

Activity measurement can be gamed. Kellogg et al. document workers contesting and working around algorithmic control [2], and simple devices that generate mouse movement would raise active time and the composite score without any work being done; the system has no gaming detection. Conversely, reading, thinking and discussion away from the keyboard count as idle.

- The evaluation used synthetic data generated with the developers' own assumptions; results on real work patterns may differ.
- The LLM components (assistant LLM mode, screenshot analysis, camera vision mode) were not evaluated with a live model.
- No usability, comprehension or well-being study with managers or employees was conducted.
- Commercial monitoring products were not compared, so the novelty claim is relative to the academic literature reviewed.
- Performance was measured on a single host with network latency excluded; the in-memory rate limiter does not coordinate across multiple server instances.
- The composite score weights (0.30/0.30/0.25/0.15) were chosen by design, not derived empirically, and the productive-application categories are organisational judgements.
- Signing in to the website starts tracking, and activity consent is mandatory to use the agent, so use of the task features is coupled to being tracked (Section 7).
- The desktop agent's operating-system hooks were not executed in this study; only its pure modules were tested.

Threats to Validity

Internal validity. The authors developed the system and also designed and ran its evaluation, including the black-box tests, benchmark and perturbation cases; the harness scenarios were written during development. Tests may therefore reflect the developers' expectations, and no independent evaluator was involved. **Construct validity.** Indicators such as active ratio measure presence at a device, not performance; the study makes no claim that they measure productivity, but readers of dashboards may interpret them that way. **Statistical conclusion validity.** The z-score detector uses population standard deviation over short windows; with seven daily points the largest attainable $|z|$ is $\sqrt{6} \approx 2.45$, so a threshold of 2.0 sits close to the mathematical limit of the method and a single inactive day can dominate it, as discussed for point-outlier techniques by Blázquez-García et al. [17]. **External validity.** Four synthetic employees over 14 days cannot represent organisational diversity, and latency on two vCPUs with a local database is not production latency. **Reliability.** The seed is deterministic for a given run date and the scripts are supplied, so the measurements can be repeated, although absolute timings will vary with hardware and the weekly windows depend on the run date.

Ethical, Privacy, and Security Considerations

Workplace monitoring raises ethical questions that software can mitigate but not resolve. Ball recommends that monitoring be proportionate, minimized and developed with worker participation [3]; König emphasises transparency and worker agency throughout the monitoring life-cycle [11];

and XAI research links explainability to accountability [14]. This section distinguishes between the privacy principles the framework intends and the controls actually implemented and verified.

Table 16: Intended principles versus implemented controls

Principle	Implemented control	Verification status
Informed awareness	Installer monitoring notice (Windows, macOS); three-step in-app consent wizard; per-screenshot notification; visible tray icon	Code reviewed; GUI not executed in this study
Consent and withdrawal	Per-type consent stored and enforced by server; withdrawal stops tracking and deletes screenshots; reconsent on terms change (v1 → v2). Activity-time consent is mandatory for using the agent, and web sign-in starts tracking	Server tests; ingestion tests. Voluntariness limited (see below)
Purpose limitation	Data used for dashboards, reports and assistant only; no export or third-party sharing feature	Code reviewed; organisational use not controllable
Data minimisation	Counts and durations only; no keystroke content, titles or URLs; unknown fields not stored; assistant LLM receives no descriptions, comments or images	Schemas, code and E4 test
Retention of raw data	TTL 90 days for raw activity and application usage; camera labels 30 days (default); screenshots deleted after 3 days (hourly job)	Indexes and job verified in code; long-duration expiry not observed
Retention of derived and audit data	No retention limit for WorkSession, AgentRun (questions, fact sheets with names and task titles), DailyAnalysis (kept after screenshots are deleted), Report and Performance	Gap: indefinite retention
Access control and accountability	Role and team scoping; 404 for out-of-scope; screenshot view log visible to employee; AgentRun audit log	Server tests; F07–F11
Human oversight	Manager approval of tasks; AI output labelled and evidence-linked; no automated decisions	F15–F21; harness
Employee rights (access, correction, erasure)	Employee sees own data, deletes own screenshots; no self-service export or erasure of activity records	Partial
Bias and fairness	No demographic data used; equal formula for all employees	Not assessed empirically (e.g., role or work-style differences)

Voluntariness of consent. Activity-time consent is required to use the desktop agent, and signing in to the website sets the tracking state to active. An employee who wishes to use the task features without being tracked has no option other than pausing manually. In an employment relationship this weakens the meaning of consent; decoupling dashboard sign-in from tracking, and making

tracking a separate, revocable choice, would align the implementation more closely with the stated principle.

Table 17: Data sent to the external LLM provider (optional features)

Feature	Data sent	Risk	Mitigation in code / recommended
AI assistant (LLM mode)	Employee names and codes, task titles, dates, counts (fact sheet)	Personal data processed by a third party, possibly outside India	No descriptions, comments or images sent; recommended: pseudonymise names, data-processing agreement
Analysis agent	One consented screenshot at a time	Screenshots may contain third-party or confidential content	Generic labels only; images retained 3 days; recommended: on-device classification option
Camera agent (vision mode)	One webcam frame per check	Biometric-adjacent imagery; bystanders	Local mode by default; separate vision consent; label must exclude identity and emotion

Legal frameworks. In the European Union, the General Data Protection Regulation (Regulation (EU) 2016/679) applies to the processing of employees' personal data and sets out principles including lawfulness, purpose limitation, data minimisation and storage limitation [25]. In India, the Digital Personal Data Protection Act, 2023 [26] establishes obligations for data fiduciaries; the Digital Personal Data Protection Rules, 2025 were notified on 13 November 2025 with phased commencement, the notice and security-safeguard provisions taking effect from 13 May 2027 [27]. WorkPlus implements technical measures that are relevant to these frameworks—consent records, minimisation, retention limits for raw data and access control—but the authors make no claim of regulatory compliance. The indefinite retention of derived records and the transfer of personal data to an external LLM provider would both need to be addressed by a deploying organisation, together with its lawful basis, notices, data-processing agreements and procedures for data-principal requests. Under GDPR, the voluntariness of consent in an employment relationship is itself contested, so consent should not be the only safeguard.

Responsible AI. The NIST AI Risk Management Framework organises AI risk management into the functions Govern, Map, Measure and Manage [24]. Mapped onto WorkPlus, the design contributes to Map (documented purpose and data flows), Measure (the evaluation harness and audit log) and Manage (fallbacks and removal of unsupported statements), while Govern—policies, roles and accountability for how outputs are used—lies with the deploying organisation. The report defects identified in Section 5.5 show why continued measurement matters: a correct-looking report can still encode a wrong assumption.

Psychological risks. Even transparent monitoring is associated with stress [10]. Pause controls, the absence of an AI-generated effort score in the assistant and supportive recommendation wording reduce but do not remove this risk. Optional screenshots and camera checks increase invasiveness; although they are consent-gated, short-retention and label-only (camera), organisations should justify them case by case.

Security and misuse. Implemented controls are described in Section 4.8. Residual risks include: the shared service key for internal routes; long-lived

agent tokens (30 days); dependence on third-party hosting and an external LLM provider for optional features; the absence of multi-factor authentication and email-based recovery; screenshot storage on ephemeral disk in the default free deployment; and open registration, which allows anyone to create an organisation and deploy monitoring without any check of purpose. No penetration test or assessment against an information-security standard was performed.

X. CONCLUSION AND FUTURE WORK

This paper presented WorkPlus, a privacy-aware employee performance and activity tracking system, and evaluated it as a design science artefact. The system combines consent-gated, content-free activity measurement with a manager-reviewed task lifecycle and an AI-assisted reporting pipeline in which the server computes, the model interprets and the server verifies numbers and citations. Under controlled, developer-run conditions with synthetic data, all automated tests, hand-computed metric checks and black-box API tests passed, server-side access controls behaved as specified, and local response times were in the low tens of milliseconds. The evaluation also characterised the limits of the grounding check and uncovered two report defects: inactive days distort trends and anomalies, and scores are aggregated differently in different components. The evidence still missing is an evaluation of the LLM path, studies with real users, and data from real work.

Meta-analytic evidence indicates that monitoring does not by itself improve performance [10]. The contribution of this work is therefore not a claim that tracking improves work, but an architecture that makes it harder for activity data and AI narratives to be mistaken for evidence of performance, together with an account of where it currently falls short. Field evidence on AI-assisted knowledge work shows benefits within, and errors outside, the capability frontier [16]; grounding and labelling are intended to keep AI-assisted reporting on the safer side of that frontier, but Section 5.5 shows that they do not yet do so for qualitative claims.

Future work, in order of priority

- Exclude inactive days from trend and anomaly series (with a working-day calendar and minimum-window rule) and use one score aggregation method across dashboard, fallback and AI service; add regression tests for both.
- Resolve the terminology tension by renaming the composite consistently and reporting activity factors and task-outcome rates separately.
- Run harness Part C with a live LLM, add human attribution judgements and usefulness ratings, and extend the check to numbers written as words.
- Define retention periods for derived and audit records, decouple dashboard sign-in from tracking, and pseudonymise names sent to the LLM.
- Conduct an ethics-approved field study with consenting employees, evaluated independently of the developers, measuring comprehension of indicators, perceived fairness and stress with validated instruments.
- Add gaming-resistance analysis, multi-instance rate limiting, multi-factor authentication and an assessment against recognised information-security and privacy standards.

REFERENCES

1. Yang, L., Holtz, D., Jaffe, S., Suri, S., Sinha, S., Weston, J., Joyce, C., Shah, N., Sherman, K., Hecht, B. and Teevan, J. (2022) 'The effects of remote work on collaboration among information workers', *Nature Human Behaviour*, 6(1), pp. 43–54. doi: 10.1038/s41562-021-01196-4.
2. Kellogg, K. C., Valentine, M. A. and Christin, A. (2020) 'Algorithms at Work: The New Contested Terrain of Control', *Academy of Management Annals*, 14(1), pp. 366–410. doi: 10.5465/annals.2018.0174.
3. Ball, K. (2021) *Electronic Monitoring and Surveillance in the Workplace: Literature review and policy recommendations*. Luxembourg: Publications Office of the European Union, JRC125716. ISBN 978-92-76-43340-8. doi: 10.2760/5137.

4. Siegel, R., König, C. J. and Lazar, V. (2022) 'The impact of electronic monitoring on employees' job satisfaction, stress, performance, and counterproductive work behavior: A meta-analysis', *Computers in Human Behavior Reports*, 8, p. 100227. doi: 10.1016/j.chbr.2022.100227.
5. Madhumita, G., Dolly Diana, P., Neena, P. C., Narendra Kiran, P. B., Aggarwal, S. and Nargunde, A. S. (2024) 'AI-powered Performance Management: Driving Employee Success and Organizational Growth', in *2024 5th International Conference on Recent Trends in Computer Science and Technology (ICRTCST)*. Jamshedpur, India: IEEE, pp. 204–209. doi: 10.1109/ICRTCST61793.2024.10578371.
6. Suvedhaa, A., Imranmohamed, M., Sudhipth, V. and Tamilarasi, M. (2025) 'AI-Powered Workforce Management for Automated Appraisals and Task Allocation', in *2025 6th International Conference on Innovative Trends in Information Technology (ICITIIT)*. Kottayam, India: IEEE. doi: 10.1109/ICITIIT64777.2025.11041265.
7. Duggan, J., Sherman, U., Carbery, R. and McDonnell, A. (2020) 'Algorithmic management and appwork in the gig economy: A research agenda for employment relations and HRM', *Human Resource Management Journal*, 30(1), pp. 114–132. doi: 10.1111/1748-8583.12258.
8. Jarrahi, M. H., Newlands, G., Lee, M. K., Wolf, C. T., Kinder, E. and Sutherland, W. (2021) 'Algorithmic management in a work context', *Big Data & Society*, 8(2). doi: 10.1177/20539517211020332.
9. Ravid, D. M., Tomczak, D. L., White, J. C. and Behrend, T. S. (2020) 'EPM 20/20: A Review, Framework, and Research Agenda for Electronic Performance Monitoring', *Journal of Management*, 46(1), pp. 100–126. doi: 10.1177/0149206319869435.
10. Ravid, D. M., White, J. C., Tomczak, D. L., Miles, A. F. and Behrend, T. S. (2023) 'A meta-analysis of the effects of electronic performance monitoring on work outcomes', *Personnel Psychology*, 76(1), pp. 5–40. doi: 10.1111/peps.12514.
11. König, C. J. (2025) 'Electronic Monitoring at Work', *Annual Review of Organizational Psychology and Organizational Behavior*, 12(1), pp. 321–342. doi: 10.1146/annurev-orgpsych-110622-060758.
12. Tambe, P., Cappelli, P. and Yakubovich, V. (2019) 'Artificial Intelligence in Human Resources Management: Challenges and a Path Forward', *California Management Review*, 61(4), pp. 15–42. doi: 10.1177/0008125619867910.
13. Forsgren, N., Storey, M.-A., Maddila, C., Zimmermann, T., Houck, B. and Butler, J. (2021) 'The SPACE of developer productivity', *Communications of the ACM*, 64(6), pp. 46–53. doi: 10.1145/3453928.
14. Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., Chatila, R. and Herrera, F. (2020) 'Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI', *Information Fusion*, 58, pp. 82–115. doi: 10.1016/j.inffus.2019.12.012.
15. Parent-Rocheleau, X. and Parker, S. K. (2022) 'Algorithms as work designers: How algorithmic management influences the design of jobs', *Human Resource Management Review*, 32(3), p. 100838. doi: 10.1016/j.hrmr.2021.100838.
16. Dell'Acqua, F., McFowland III, E., Mollick, E., Lifshitz, H., Kellogg, K. C., Rajendran, S., Krayer, L., Candelon, F. and Lakhani, K. R. (2026) 'Navigating the Jagged Technological Frontier: Field Experimental Evidence of the Effects of Artificial Intelligence on Knowledge Worker Productivity and Quality', *Organization Science*, Articles in Advance, pp. 1–21. doi: 10.1287/orsc.2025.21838.
17. Blázquez-García, A., Conde, A., Mori, U. and Lozano, J. A. (2021) 'A Review on Outlier/Anomaly Detection in Time Series Data', *ACM Computing Surveys*, 54(3), pp. 1–33. doi: 10.1145/3444690.
18. Amershi, S., Weld, D., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., Suh, J., Iqbal, S., Bennett, P. N., Inkpen, K., Teevan, J., Kikin-Gil, R. and Horvitz, E. (2019) 'Guidelines for Human-AI Interaction', in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Glasgow, Scotland, UK: ACM, pp. 1–13. doi: 10.1145/3290605.3300233.

19. Liao, Q. V., Gruen, D. and Miller, S. (2020) 'Questioning the AI: Informing Design Practices for Explainable AI User Experiences', in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Honolulu, HI, USA: ACM, pp. 1–15. doi: 10.1145/3313831.3376590.
20. Benlian, A., Wiener, M., Cram, W. A., Krasnova, H., Maedche, A., Möhlmann, M., Recker, J. and Remus, U. (2022) 'Algorithmic Management: Bright and Dark Sides, Practical Implications, and Research Opportunities', *Business & Information Systems Engineering*, 64(6), pp. 825–839. doi: 10.1007/s12599-022-00764-w.
21. Hevner, A. R., March, S. T., Park, J. and Ram, S. (2004) 'Design Science in Information Systems Research', *MIS Quarterly*, 28(1), pp. 75–105. doi: 10.2307/25148625.
22. Peffers, K., Tuunanen, T., Rothenberger, M. A. and Chatterjee, S. (2007) 'A Design Science Research Methodology for Information Systems Research', *Journal of Management Information Systems*, 24(3), pp. 45–77. doi: 10.2753/MIS0742-1222240302.
23. Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B. and Liu, T. (2025) 'A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions', *ACM Transactions on Information Systems*, 43(2), pp. 1–55. doi: 10.1145/3703155.
24. National Institute of Standards and Technology (2023) *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, NIST AI 100-1. Gaithersburg, MD: U.S. Department of Commerce. doi: 10.6028/NIST.AI.100-1.
25. European Parliament and Council of the European Union (2016) Regulation (EU) 2016/679 (General Data Protection Regulation), *Official Journal of the European Union*, L 119, 4 May 2016, pp. 1–88. Available at: http://data.europa.eu/eli/reg/2016/679/oj.
26. Government of India (2023) *The Digital Personal Data Protection Act, 2023* (Act No. 22 of 2023), assented 11 August 2023. New Delhi: Ministry of Law and Justice. Available at: https://www.indiacode.nic.in/bitstream/123456789/22037/2/a2023-22.pdf.
27. Ministry of Electronics and Information Technology (2025) *Digital Personal Data Protection Rules, 2025*, G.S.R. 846(E), notified 13 November 2025. New Delhi: Government of India.
28. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S. and Kiela, D. (2020) 'Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks', in *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*. Vancouver, Canada: Curran Associates. Available at: https://papers.nips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.
29. Rashkin, H., Nikolaev, V., Lamm, M., Aroyo, L., Collins, M., Das, D., Petrov, S., Tomar, G. S., Turc, I. and Reitter, D. (2023) 'Measuring Attribution in Natural Language Generation Models', *Computational Linguistics*, 49(4), pp. 777–840. doi: 10.1162/coli_a_00486.