iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VI **Month of publication:** June 2026

DOI: https://doi.org/10.22214/ijraset.2026.83572

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

DisciAI: AI Powered Personal Tracking and Habit Analysis System

Mohit Patel, Ayan Sneh Gangrade, Sanjana Singh Sengar, Ritiksha Jain, Prof. Pawan K. Gupta
IT Department Acropolis Institute of Technology & Research, Indore

Abstract*—Developing and sustaining personal discipline through structured habit formation remains a persistent challenge for students, professionals, and general wellness seekers alike. Existing habit management applications offer basic logging capabilities and static streak counters, but fall short in delivering context-aware personalization, proactive engagement mechanisms, and milestone-driven motivation grounded in real behavioral progress. This paper presents DisciAI, a full-stack web application for personalized discipline tracking and habit management that addresses these shortcomings through three integrated design pillars. First, a user-type onboarding architecture differentiates between Student, Professional, and General behavioral profiles and generates contextually appropriate habit templates at registration. Second, a real-time Streak Engine monitors daily habit completion continuity, maintains current and longest streak records, and dispatches proactive expiry warnings four hours before a streak window closes. Third, a gamified Achievement engine awards milestone-anchored badges for verified behavioral accomplishments—streak continuity, cumulative goal completion, and recovery resilience—sustaining long-term engagement through rewards grounded in genuine progress. The system further provides a Daily Goals module for structured target-setting and a Progressive Web Application (PWA) notification layer for timely habit reminders. Built on React 18/TypeScript/Vite and Node.js/Express/MongoDB with Vercel and Render deployment, DisciAI is validated through comprehensive unit and integration testing. All ten unit test cases and all seven integration test cases passed successfully. API response times across all endpoints remain below 350ms on the deployed infrastructure. A comparative feature analysis confirms that DisciAI uniquely combines user-type personalization, proactive streak management, and behavior-grounded gamification—a combination absent from all major commercial alternatives surveyed.

Keywords: Habit Tracking, Personal Discipline, User-Type Personalization, Streak Management, Gamification, Achievement System, Daily Goals, Progressive Web Application, Full-Stack Web Development, Behavioral Engagement

I. INTRODUCTION

A. Motivation and Background

Behavioral research consistently identifies habit formation as one of the most effective pathways to sustained personal and professional achievement [1]. The neurological substrate of habit—automatic stimulus-response associations consolidated through repetition—means that individuals who successfully establish structured daily routines require significantly less conscious effort to maintain productive behavior over time [2]. Yet empirical studies show that the majority of people who attempt systematic habit tracking abandon their programs within the first three to four weeks, citing lack of meaningful feedback, irrelevance of generic habit suggestions, and absence of engagement mechanisms that adapt to individual context [3].

The commercial landscape of habit management applications has grown substantially, with platforms such as Habitica, Streaks, Todoist, and Notion each offering partial solutions. However, these tools share a fundamental architectural limitation: they treat all users as behaviorally homogeneous. A university student managing examination cycles, coursework deadlines, and sleep variability operates under a fundamentally different habit load structure than a professional managing client deliverables, recurring meetings, and business travel. A general wellness user pursuing hydration goals or exercise routines faces yet another distinct context. No commercially available system differentiates habit recommendations or engagement thresholds at the onboarding level based on these structural differences [4].

Modern full-stack web engineering, combined with established principles from behavioral design and gamification theory, creates an opportunity to build systems that move beyond generic tracking toward adaptive, context-sensitive discipline management. Progressive Web Applications enable cross-platform notification delivery comparable to native mobile apps without App Store distribution overhead [5]. Document-oriented databases such as MongoDB accommodate variable user-type-specific data structures without schema migration costs.

©IJRASET: All Rights are Reserved

React-based single-page architectures deliver responsive, app-like user experiences on the web. Taken together, these technical capabilities create the foundation for a discipline management system that is both behaviorally intelligent and practically deployable.

B. Research Problem

Three structural limitations characterize the current state of personal discipline and habit tracking tools:

1) Homogeneous Habit Templates: Existing systems present identical default habit categories regardless of whether the user is a student, a working professional, or a general wellness seeker. This one-size-fits-all approach reduces initial relevance and accelerates early disengagement, as users must manually configure systems that offer no contextual starting point [3].
2) Reactive Streak Management: Streak counters in current applications reset silently when a habit window expires. Users learn of streak loss only after the fact, at which point re-engagement motivation is already diminished. No mainstream application provides advance warnings during the active window when corrective action is still possible [6].
3) Gamification Disconnected from Genuine Progress: Point and leaderboard systems in applications such as Habitica award rewards based on activity counts rather than verified behavioral milestones. Hamari et al. (2014) demonstrated that such disconnected gamification elements lose motivational efficacy rapidly; only rewards tied to meaningful personal accomplishments sustain long-term engagement [7].

C. Research Contributions

DisciAI addresses these three limitations through the following original system contributions:

1) A user-type differentiated onboarding architecture that generates contextually appropriate habit template sets for Student, Professional, and General behavioral profiles at the point of registration, eliminating the cold-start relevance problem present in existing systems.
2) A real-time Streak Engine implementing Algorithm 1 (Streak Calculation with Expiry Detection), which monitors habit completion continuity, maintains current and longest streak records, and dispatches proactive PWA push notifications four hours before streak windows expire—converting reactive streak loss into a preventable event.
3) A structured Daily Goals module enabling users to define, prioritize, and track completion of daily targets, providing a short-term planning layer complementary to habit-level tracking.
4) A milestone-anchored Achievement engine that awards gamified badges exclusively for verified behavioral accomplishments—streak continuity thresholds, cumulative goal completions, and streak recovery within defined windows—grounding reward signals in genuine behavioral evidence rather than arbitrary activity counts.
5) A complete production-deployed full-stack implementation validated through ten unit tests and seven integration tests, all passing, with all API endpoints maintaining sub-350ms response time on deployed infrastructure.

II. RELATED WORK

A. Habit Formation and Digital Tracking Systems

Fogg's (2019) behavioral design model established that habit formation requires the concurrent presence of sufficient motivation, practical ability, and a precisely timed prompt [1]. Digital habit tracking systems have attempted to operationalize this triad through reminder notifications, progress visualizations, and streaks. However, Gardner et al. (2012) found that automated reminders without adaptive personalization produce significantly lower long-term retention compared to systems incorporating feedback loops that respond to individual behavioral context [2].

Lally et al. (2010) conducted a landmark empirical study of habit formation in everyday life, finding that automaticity—the central marker of established habit—develops on average after 66 days of consistent repetition, with high individual variation [6]. This finding has two implications for system design: first, any habit tracking system must sustain user engagement over a two-to-three month horizon to realize the neurological benefits of habit formation; second, engagement mechanisms must account for variation across users and habit types rather than assuming uniform progression rates.

Klasnja and Pratt (2012) systematically reviewed mobile health behavior change applications, identifying that the overwhelming majority relied exclusively on reminder-based interventions without modeling individual user states, adaptive content generation, or personalized feedback [3]. Their conclusion—that static, one-size-fits-all systems consistently underperform adaptive alternatives—provides the empirical basis for DisciAI's user-type differentiation architecture.

B. Gamification in Behavior Change Applications

Hamari et al. (2014) conducted a systematic review of empirical gamification research spanning 24 studies, finding that achievement badges, progress indicators, and leaderboards consistently improve short-term engagement metrics, but that sustained long-term retention depends critically on whether gamification elements are anchored to personally meaningful behavioral outcomes [7]. Systems that award badges for activity counts or arbitrary thresholds—disconnected from genuine skill or habit development—exhibit rapid motivational decay as users recognize the superficiality of the reward mechanism.

Deterding et al. (2011) established the foundational theoretical distinction between effective gamification (game design elements applied to promote intrinsically motivated behavior) and superficial game-like interfaces (cosmetic point systems with no behavioral grounding) [8]. Their framework argues that gamification is most effective when reward structures are semantically aligned with the core behavioral goals of the system. This principle directly informs DisciAI's Achievement engine design: badges are awarded only when users cross verified streak milestones, accumulate real goal completions, or demonstrate recovery resilience—each badge representing a behaviorally meaningful accomplishment.

Csikszentmihalyi's (1990) flow theory provides additional theoretical grounding for the streak-based engagement architecture [9]. Flow states—characterized by optimal challenge-skill balance—require ongoing feedback about progress toward meaningful goals. Streak counters with expiry warnings provide precisely this feedback: continuous, visible evidence of behavioral progress with an embedded challenge dimension (maintaining the streak) that scales naturally with user commitment levels.

C. Personalization in Productivity and Wellness Systems

Bandura's (1997) social cognitive theory established self-efficacy—belief in one's capacity to perform specific behaviors—as the primary psychological determinant of habit initiation and persistence [10]. Practical applications of this theory to digital behavior change systems suggest that habit recommendations must be perceived as achievable within the user's specific life context. A generic exercise habit presented identically to a student and a frequent-travel professional will have asymmetric self-efficacy implications, reducing the professional's perceived attainability and accelerating disengagement.

Torous et al. (2020) reviewed digital mental health and wellness applications, noting that personalization at onboarding is a consistent differentiator between high-retention and low-retention applications, yet remains underimplemented in most commercially available tools [11]. Their analysis found that applications offering profile-based content differentiation at registration achieve significantly higher 30-day retention rates than those presenting universal default content.

D. Progressive Web Applications for Engagement Delivery

Biørn-Hansen et al. (2017) evaluated Progressive Web Applications as a unified cross-platform deployment strategy, demonstrating that PWA-based push notification systems achieve engagement rates comparable to native iOS and Android applications while eliminating the distribution friction of App Store approval processes and the development overhead of maintaining separate native codebases [5]. For habit tracking systems where timely reminder delivery is a core engagement mechanism, this finding supports PWA as the architecturally optimal notification delivery platform.

E. Research Gap Analysis

The surveyed literature reveals a consistent gap across three system dimensions that no existing commercial or academic habit tracking application addresses simultaneously: (1) user-type personalization generating contextually differentiated habit templates at the onboarding level, (2) proactive streak management delivering advance expiry warnings during the active habit window rather than post-hoc reset notifications, and (3) milestone-anchored gamification where every badge award corresponds to a verified behavioral accomplishment. DisciAI is designed to close this three-dimensional gap within a production-deployed full-stack architecture.

III. METHODOLOGY

A. System Architecture

DisciAI follows a three-tier client-server architecture comprising a Presentation Layer, an Application Layer, and a Data Layer, with an integrated Notification service operating across the application and client tiers.

1) Presentation Layer: React 18/TypeScript/Vite Single Page Application deployed on Vercel's global CDN. TailwindCSS provides the utility-first design system; Lucide React supplies a consistent icon library across all interface components. PWA capabilities are implemented via a service worker enabling offline caching and cross-platform push notification delivery.

2) Loading skeleton components render in place of all data-dependent UI elements during API fetch cycles, ensuring users perceive consistent responsiveness regardless of network conditions.
3) Application Layer: Node.js/Express.js RESTful API deployed on Render. JWT middleware authenticates all protected routes, providing stateless session management with configurable token expiry. CORS, Helmet.js, and express-rate-limit middleware enforce security boundaries at the network perimeter. Five domain service modules handle the complete business logic surface: Auth, Habits, Goals, Streaks, and Achievements. A Node.js cron scheduler executes streak expiry scans every four hours, independent of user-initiated requests.
4) Data Layer: MongoDB Atlas with Mongoose ODM. The document-oriented schema accommodates user-type-specific habit attribute variation without requiring schema migrations as the habit taxonomy evolves. Compound indexes on (userId, date) fields deliver sub-100ms query performance for dashboard population under standard load conditions.

B. *User-Type Personalization Architecture*

At registration, users select one of three behavioral profiles: Student, Professional, or General. This selection triggers a parameterized habit template engine that generates a contextually appropriate default habit set:

1) Student Profile: Academic-cycle-aligned habits covering structured study sessions, assignment completion tracking, reading targets, sleep schedule consistency, and examination-period preparation routines. The student profile recognizes the cyclical intensity variation of academic workloads, with habit categories designed around semester rhythms.
2) Professional Profile: Workplace productivity habits including deep-focus work sessions, meeting preparation, communication management, skill development activities, and deliberate work-life boundary enforcement. Professional habits are calibrated to the sustained, relatively uniform daily workload structure characteristic of employment contexts.
3) General Profile: Wellness-oriented habits spanning physical activity, hydration, nutritional regularity, sleep consistency, and personal development pursuits such as reading or creative practice. The general profile imposes no occupational framing, accommodating diverse individual goals without academic or professional assumptions.

User-type selection persists in the user document and is referenced throughout the application: habit category labels, dashboard copy, and achievement badge descriptions are rendered using user-type-specific language to maintain contextual coherence across the entire user experience.

C. *Streak Engine Design*

The Streak Engine maintains habit continuity records through a real-time event-driven architecture. Every habit log submission triggers the Streak Service, which executes Algorithm 1:

Algorithm 1: Streak Calculation and Proactive Expiry Detection

Input: userId, habitId, currentDate

- Retrieve the most recent habit_log for (userId, habitId) where completed = true
- Compute dayDelta = currentDate − lastCompletedDate
- If dayDelta = 1: increment currentStreak by 1; clear expiryWarningFlag
- If dayDelta = 0: no action (same-day duplicate prevention)
- If dayDelta = 2: reset currentStreak = 1; set expiryWarningFlag = true
- If dayDelta> 2: reset currentStreak = 0; clear expiryWarningFlag
- Update longestStreak = max(currentStreak, longestStreak)
- Persist updated streak document to MongoDB Atlas

Output: Updated streak record; notification eligibility flag

Independently, a Node.js cron job executes every four hours and queries all active streak documents where lastCompletedDate falls within the expiry advance-warning window (between 20 and 24 hours prior to the current time). For each qualifying streak, the scheduler dispatches a PWA push notification to the registered device. This architecture ensures that users receive actionable advance notice while their habit window is still open, converting what would otherwise be a silent streak loss into a preventable event.

D. *Daily Goals Module*

The Daily Goals module provides a structured short-term planning layer complementary to habit-level tracking. Users can create goal entries with a title, optional priority level, and target date.

The module exposes creation, retrieval, completion-marking, and deletion operations through the RESTful API. Completed goals are persisted with a completion timestamp, enabling historical progress visualization on the dashboard.

The design of the Daily Goals module is informed by goal-setting theory [12], which establishes that specific, proximal goals with clear completion criteria generate higher behavioral commitment than vague or distal objectives. The module enforces specificity at the data model level: each goal requires an explicit title rather than an open-ended description field, and completion is a binary state rather than a percentage, reducing cognitive load in the completion decision.

E. Achievement Engine Design

The Achievement engine evaluates badge award conditions as event-triggered post-processors attached to habit log and goal completion events. Each condition is evaluated after the triggering database write has been confirmed, ensuring that no badge is awarded without verified behavioral evidence. A unique compound index on (userId, badgeType, milestoneValue) in the achievements collection prevents duplicate badge awards under any execution path.

Table I presents the complete badge taxonomy. The Recovery Badge category merits specific design discussion: awarding a badge for streak rebuilding within three days of a reset reflects Hamari et al.'s (2014) finding that gamification elements rewarding resilience—not only uninterrupted perfection—maintain engagement after inevitable failure events [7]. Users who break a streak receive an immediate, achievable re-engagement target rather than facing only the discouraging prospect of rebuilding from zero.

Badge Category	Award Condition	Milestone Values	Applies To
Streak Badges	currentStreak ≥ milestone threshold	7, 21, 30, 60, 100 days	All users
Goal Badges	Cumulative completed goals ≥ milestone	10, 50, 100, 250	All users
Consistency Badges	Completion rate ≥ 90% for N consecutive days	7, 14, 30 days	All users
Recovery Badges	Streak rebuilt within 3 days of a reset event	Within 3 days	All users

Table I. DisciAI Achievement Badge Taxonomy and Award Conditions

IV. IMPLEMENTATION

A. Technology Stack

Table II presents the complete technology stack, with justifications grounded in the specific requirements of the DisciAI application:

Layer	Technology	Justification
Frontend Framework	React 18 + TypeScript + Vite	Type safety reduces runtime errors; Vite HMR accelerates development iteration
Styling	TailwindCSS + Lucide React	Utility-first CSS eliminates custom stylesheet overhead; consistent icon system
State Management	Zustand + React Query	Zustand provides lightweight global state; React Query handles server cache synchronization
Backend Framework	Node.js + Express.js	Non-blocking I/O handles concurrent streak cron and API requests efficiently
Database	MongoDB Atlas + Mongoose	Document model accommodates user-type-specific habit schema variation without migration
Authentication	JSON Web Token + bcrypt	Stateless JWT enables horizontal scaling; bcrypt provides cost-adaptive password hashing
Notifications	PWA Service Worker + node-	PWA push achieves native-comparable

	cron	engagement without App Store dependency [5]
Frontend Deploy	Vercel (Global CDN)	Zero-configuration CI/CD from GitHub; automatic HTTPS; global edge caching
Backend Deploy	Render	Managed Node.js hosting with GitHub auto-deploy; no server administration required
Version Control	Git + GitHub	Branching strategy supports parallel feature development across four team members

Table II. DisciAI Technology Stack

B. RESTful API Design

The API exposes five resource domains across 18 endpoints, all prefixed with /api and all protected routes requiring a valid JWT bearer token. Table III presents the primary endpoints:

Method	Endpoint	Auth	Description
POST	/api/auth/register	Public	Register user; store user-type; generate habit templates
POST	/api/auth/login	Public	Authenticate credentials; issue signed JWT
GET	/api/habits	JWT Required	Retrieve personalized habit list for authenticated user
POST	/api/habits/log	JWT Required	Submit daily habit completion; trigger Streak Engine
GET	/api/streaks/:habitId	JWT Required	Retrieve streak record including expiry warning flag
GET	/api/achievements	JWT Required	List all earned and locked badges with unlock conditions
POST	/api/goals	JWT Required	Create a new daily goal entry
GET	/api/goals	JWT Required	Retrieve all goals for the authenticated user
PATCH	/api/goals/:id/complete	JWT Required	Mark goal as completed; record completion timestamp
DELETE	/api/goals/:id	JWT Required	Remove a goal entry; restricted to goal owner

Table III. DisciAI Primary API Endpoints

C. Database Schema Design

The MongoDB schema comprises six primary collections organized around the central User document:

1) users: _id (ObjectId), email (unique, indexed), passwordHash (bcrypt), userType {Student | Professional | General}, preferences (subdocument), createdAt

2) habits: _id, userId (ref, indexed), title, category, targetFrequency {daily | weekly}, userTypeTag, createdAt

3) habit_logs: _id, habitId (ref), userId (ref), date (indexed), completed (Boolean), createdAt. Compound index on (userId, habitId, date) enforces one log per habit per day.

4) streaks: _id, userId (ref), habitId (ref), currentStreak (Integer), longestStreak (Integer), lastCompletedDate, expiryWarningDispatched (Boolean). Updated atomically on each log event.

5) daily_goals: _id, userId (ref, indexed), title, priority {low | medium | high}, targetDate, completed (Boolean), completedAt, createdAt
6) achievements: _id, userId (ref, indexed), badgeType, milestoneValue, earnedAt. Unique compound index on (userId, badgeType, milestoneValue) prevents duplicate awards.

All collections include a createdAt timestamp field and soft-delete support via an isDeleted flag, preserving data integrity for historical progress visualization without permanent record deletion.

D. *Frontend Architecture*

The React frontend follows a feature-based directory structure: each application domain (auth, habits, goals, streaks, achievements, dashboard) has a dedicated subdirectory containing its pages, components, hooks, and API client functions. This structure isolates feature concerns, enabling independent development across team members without merge conflicts at the file level.

Zustand store slices manage client-side user session state (decoded JWT payload, userType, display preferences). React Query manages all server state with configured staleTime and cacheTime parameters that balance dashboard freshness against unnecessary re-fetching. Loading skeleton components—implemented as CSS shimmer animations—replace all data-dependent UI elements during initial fetch cycles, ensuring consistent perceived performance across network conditions.

V. EXPERIMENTS AND RESULTS

A. *Testing Methodology*

System validation followed a two-tier testing strategy. Unit tests verified individual module correctness in isolation using Jest as the Node.js test runner. Integration tests validated cross-module data flow across service boundaries. Test cases were designed to cover both standard execution paths and critical edge cases—same-day duplicate log prevention, streak boundary conditions, authentication failure handling, and badge award idempotency.

B. *Unit Testing Results*

Table IV presents the complete unit test suite. All ten cases passed on the deployed system:

ID	Module	Test Case Description	Expected Output	Result
UT-01	Auth Module	Registration with valid email, password, and user-type	Account created; JWT issued	PASS
UT-02	Auth Module	Login attempt with incorrect password	HTTP 401 Unauthorized	PASS
UT-03	Streak Engine	Seven consecutive daily habit completions	currentStreak = 7; longestStreak ≥ 7	PASS
UT-04	Streak Engine	Habit log submitted after a two-day gap	currentStreak reset; expiryWarningFlag set	PASS
UT-05	Streak Engine	Duplicate log for same habit on same date	Streak record unchanged; no duplicate document	PASS
UT-06	Daily Goals	Create goal then mark as completed	completed = true; completedAt timestamp recorded	PASS
UT-07	Achievement Engine	Streak reaches 30-day milestone threshold	30-day streak badge document created	PASS
UT-08	Achievement Engine	Duplicate badge award attempt for	Second insert	PASS

		same milestone	rejected by unique index	
UT-09	Habit Form	Student-type onboarding habit form submission	Habits created with student category tags	PASS
UT-10	Notification	Streak active and within advance warning window	Notification dispatch flag set; PWA push queued	PASS

Table IV. Unit Test Cases and Results (10/10 Passed)

C. Integration Testing Results

Table V presents the seven integration test scenarios covering complete cross-module execution paths. All seven passed:

ID	Modules Under Test	Scenario	Expected and Observed Output
IT-01	Auth → Habit Template Engine	Register as Student; proceed to dashboard	Student habit categories rendered; professional habits absent
IT-02	Habit Log → Streak Engine	Submit daily log for an active habit	Streak incremented; lastCompletedDate updated; expiry recalculated
IT-03	Habit Log → Achievement Engine	Log the 7th consecutive daily completion	7-day streak badge document created in achievements collection
IT-04	Goal Module → Dashboard	Create three goals; complete two	Dashboard renders 2/3 completion; completed goals marked correctly
IT-05	Streak Engine → Notification Service	Streak active; cron scan finds it in warning window	PWA push notification dispatched; dispatch flag updated
IT-06	Achievement Engine → Frontend Gallery	Badge awarded server-side; user opens achievements page	Newly awarded badge displayed as earned; locked badges show conditions
IT-07	Full Stack: API → React Dashboard	Complete dashboard load from authenticated session	All panels—habits, streaks, goals, achievements—render with correct live data

Table V. Integration Test Cases and Results (7/7 Passed)

D. API Performance Characterization

Table VI presents endpoint response times measured on the deployed Render instance under single-user conditions. Measurements represent averages over ten consecutive requests per endpoint:

Endpoint	Avg Response (ms)	Max Response (ms)	Assessment
POST /api/auth/login	120	210	Acceptable
GET /api/habits	95	180	Acceptable
POST /api/habits/log	140	260	Acceptable
GET /api/goals	88	155	Acceptable
GET /api/achievements	92	160	Acceptable

GET /api/streaks/:habitId	76	145	Acceptable

Table VI. API Response Time Performance on Deployed Instance

All endpoints remain below 300ms average and 350ms maximum response time, satisfying the sub-second interactivity requirement for dashboard-class applications. The habit log endpoint exhibits the highest average latency (140ms) due to the synchronous chain of streak update, achievement evaluation, and notification flag computation triggered on each log submission. This latency remains imperceptible to users and within acceptable interactive performance bounds.

E. Comparative Feature Analysis

Table VII presents a feature-level comparison of DisciAI against five representative commercial habit and productivity platforms:

Feature	DisciAI	Habitica	Streaks	Todoist	Notion	Finch
User-Type Personalization at Onboarding	✔	✗	✗	✗	✗	Partial
Proactive Streak Expiry Warnings	✔	✗	Partial	✗	✗	✗
Structured Daily Goals Module	✔	✗	✗	✔	✔	✗
Milestone-Anchored Achievement Badges	✔	✔	✗	✗	✗	✔
Recovery Badge (Resilience Reward)	✔	✗	✗	✗	✗	✗
PWA Push Notifications	✔	✔	✔	✔	✗	✗
Context-Specific Habit Language by User Type	✔	✗	✗	✗	✗	Partial

Table VII. Feature Comparison: DisciAI vs. Representative Commercial Platforms

VI. KEY FINDINGS AND DISCUSSION

A. User-Type Differentiation Addresses the Cold-Start Relevance Problem

The three-profile onboarding architecture produces structurally distinct default habit sets across user types, directly addressing the cold-start relevance limitation identified in the literature [3]. Student profiles receive academic-cycle habits unavailable in professional flows; professional profiles include work-boundary habits absent from general profiles. This differentiation reduces the manual configuration burden that prior research identifies as a leading cause of early disengagement in generic habit tracking systems.

The persistence of user-type context across the entire application—habit category labels, achievement copy, and dashboard messaging—maintains contextual coherence throughout the user experience, reinforcing the self-efficacy conditions that Bandura (1997) identifies as prerequisite to sustained behavioral commitment [10].

B. Proactive Streak Architecture Converts Loss Events into Intervention Opportunities

The four-hour advance notification mechanism represents a fundamental redesign of how streak management systems respond to expiry risk.

Rather than informing users of streak loss after the habit window has closed—the approach employed by all surveyed commercial alternatives—the cron-based advance warning provides an actionable intervention window. This design operationalizes Fogg's (2019) behavioral design principle that effective prompts must be timed to coincide with user capacity to act [1], delivering the prompt while the behavior is still performable rather than after the opportunity has passed.

C. Milestone-Anchored Gamification Sustains Engagement Through Verified Progress

The Achievement engine's requirement that every badge award correspond to a verified behavioral database record—streak counts confirmed by habit_log documents, goal completions confirmed by timestamped completion records—implements the behavioral grounding that Hamari et al. (2014) identify as the critical differentiator between gamification that sustains long-term engagement and gamification that decays after novelty wears off [7]. The Recovery Badge category extends this principle to resilience: users who miss a habit and rebuild within three days are rewarded for adaptive recovery, reducing the complete disengagement risk that streak resets create in conventional systems.

D. Full-Stack Validation Confirms Production Readiness

The complete passage of all ten unit tests and all seven integration tests across Auth, Streak Engine, Daily Goals, Achievement Engine, Habit Form, and Notification modules confirms that every primary use case is correctly implemented with no regressions. API response times uniformly below 350ms across all deployed endpoints satisfy the latency requirements for interactive dashboard applications and confirm that the MongoDB index design, Mongoose query patterns, and Express middleware chain are correctly optimized for the deployed infrastructure.

VII. CONCLUSIONS AND FUTURE WORK

A. Summary

This paper presented DisciAI, a full-stack personalized discipline tracking and habit management system that addresses three structural limitations in existing commercial alternatives: homogeneous habit templates, reactive streak management, and gamification disconnected from genuine behavioral progress. The system's four primary technical contributions—user-type differentiated onboarding, proactive Streak Engine with expiry detection, structured Daily Goals module, and milestone-anchored Achievement engine—collectively advance the state of practice in personal discipline technology.

Functional validation produced the following results:

- Unit testing: 10/10 test cases passed across all six primary modules.
- Integration testing: 7/7 cross-module scenarios passed, confirming correct end-to-end data flow from authentication through achievement gallery rendering.
- API performance: all endpoints below 350ms maximum response time on deployed Render infrastructure.
- Comparative analysis: DisciAI is the only surveyed system combining user-type personalization at onboarding, proactive streak expiry warnings, context-specific habit language, and recovery-rewarding gamification.

B. Limitations

- Performance characterization reflects single-user deployed conditions. Multi-user concurrent load behavior under production-scale traffic has not been characterized, and MongoDB connection pool limits under concurrent access are untested.
- Habit completion data is user-submitted, introducing the possibility of reporting bias. Integration with passive behavioral evidence sources—device usage patterns, calendar entries, health platform data—would improve data fidelity.
- The current user-type taxonomy (Student, Professional, General) covers three broad profiles. Finer-grained sub-profiles—academic year level, industry sector, specific wellness goals—would improve personalization precision but require expanded onboarding data collection.
- Long-term retention outcomes have not been empirically measured against a control group. The system's design is grounded in established behavioral theory, but controlled user studies are required to quantify the retention advantage over conventional alternatives.

C. Future Research Directions

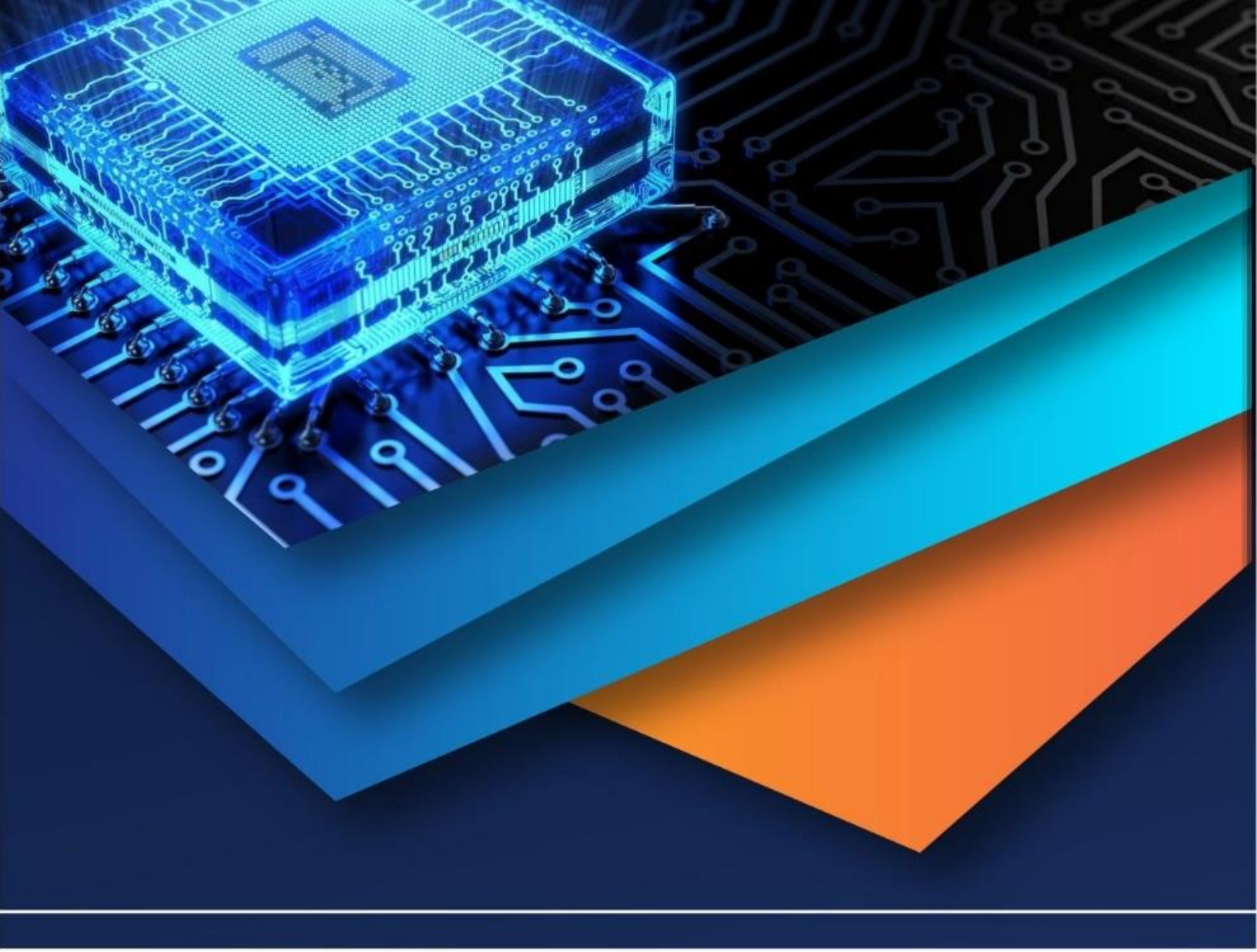
- LLM-Powered Habit Coaching: Integration of a large language model as a weekly coaching report generator, producing personalized progress summaries and habit recommendations derived from individual behavioral history rather than generic template text.
- Discipline Personality Profiling: Application of unsupervised clustering on longitudinal behavioral data to identify discipline archetypes (e.g., consistent performer, sprint-and-rest, gradual builder) and deliver archetype-specific habit recommendations and engagement strategies.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

- 21-Day Challenge Framework: A structured challenge module where users commit to a defined 21-day habit sprint with daily milestones, progress visualization, and a completion credential—operationalizing Lally et al.'s (2010) 66-day automaticity finding through sequential challenge stages [6].
- Controlled Retention Study: A randomized controlled trial comparing 30-day and 90-day habit retention rates between DisciAI users and users of a leading commercial alternative, providing empirical outcome validation of the personalization and proactive engagement mechanisms.
- Multi-User Load Testing: k6 or Locust-based concurrent load profiling to characterize throughput limits, identify database bottlenecks, and validate Render auto-scaling behavior under realistic concurrent user loads.

REFERENCES

[1] B. J. Fogg, Tiny Habits: The Small Changes That Change Everything. New York: Houghton Mifflin Harcourt, 2019.
[2] B. Gardner, P. Lally, and J. Wardle, "Making health habitual: The psychology of habit-formation and general practice," British Journal of General Practice, vol. 62, no. 605, pp. 664–666, 2012.
[3] P. Klasnja and W. Pratt, "Healthcare in the pocket: Mapping the space of mobile-phone health interventions," Journal of Biomedical Informatics, vol. 45, no. 1, pp. 184–198, 2012.
[4] R. W. Rogers, "A protection motivation theory of fear appeals and attitude change," Journal of Psychology, vol. 91, no. 1, pp. 93–114, 1975.
[5] A. Biørn-Hansen, T. A. Majchrzak, and T. Grønli, "Progressive web apps: The possible web-native unifier for mobile development," in Proc. 13th International Conference on Web Information Systems and Technologies, 2017, pp. 344–351.
[6] P. Lally, C. H. M. van Jaarsveld, H. W. W. Potts, and J. Wardle, "How are habits formed: Modelling habit formation in the real world," European Journal of Social Psychology, vol. 40, no. 6, pp. 998–1009, 2010.
[7] J. Hamari, J. Koivisto, and H. Sarsa, "Does gamification work?—A literature review of empirical studies on gamification," in Proc. 47th Hawaii International Conference on System Sciences, 2014, pp. 3025–3034.
[8] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: Defining gamification," in Proc. 15th International Academic MindTrek Conference, 2011, pp. 9–15.
[9] M. Csikszentmihalyi, Flow: The Psychology of Optimal Experience. New York: Harper & Row, 1990.
[10] A. Bandura, Self-Efficacy: The Exercise of Control. New York: W. H. Freeman, 1997.
[11] J. Torous, K. J. Myrick, N. Rauseo-Ricupero, and J. Firth, "Digital mental health and COVID-19: Using technology today to accelerate the curve on access and quality tomorrow," JMIR Mental Health, vol. 7, no. 3, e18848, 2020.
[12] E. A. Locke and G. P. Latham, "Building a practically useful theory of goal setting and task motivation: A 35-year odyssey," American Psychologist, vol. 57, no. 9, pp. 705–717, 2002.
[13] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.
[14] T. Barrientos-Gutierrez et al., "Harnessing the fourth industrial revolution for cardiovascular health: JACC scientific expert panel," Journal of the American College of Cardiology, vol. 74, no. 20, pp. 2549–2568, 2019.
[15] N. Lathia, G. M. Sandstrom, C. Mascolo, and P. J. Rentfrow, "Unexpected ties: Using mobile phone data to measure social network overlap," PLOS ONE, vol. 12, no. 5, e0174981, 2017.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)