Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 6, 2026

SPACEHIVE 360: THE AI- POWERED SAAS PLATFORM FOR SMART, SEAMLESS CO- WORKING MANAGEMENT

[*1]Dr. Bushra Fazal Khan, [2]Dr. Muhammad Ashraf, [3]Hira Faisal, [4]Hamza Faisal, [5]Jawad Ahmed

[1]Assistant Professor, Department of Software Engineering, Bahria University Karachi Campus
[2]Associate Professor, Department of Computer Engineering, BUITEMS Quetta
[3]Student, Department of Software Engineering Bahria University Karachi Campus, Pakistan
[4]Student, Department of Software Engineering Bahria University Karachi Campus, Pakistan
[5]Student, Department of Software Engineering Bahria University Karachi Campus, Pakistan
[*1]bushrafazal.bukc@bahria.edu.pk, [2]muhammad.ashraf@buitms.edu.pk,
[3]hira.faisal7676@gmail.com, [4]hamza.faisal7676@gmail.com, [5]jawadahmed81958@gmail.com

DOI:-

Keywords
Coworking Management System, SaaS Platform, Artificial Intelligence, Dynamic Pricing, Sentiment Analysis, Workspace Concierge, Multi-Tenant Architecture, Recommendation System.

Article History
Received: 30 May, 2026
Accepted: 22 June, 2026
Published: 24 June, 2026

Corresponding Author: *
Dr. Bushra Fazal Khan

Abstract
One reason coworking keeps expanding? Still no central system linking different workspace operators together digitally. Instead, most current setups either connect locations physically or handle just one provider's operations - neither handles smart automation well across large volumes. Enter SpaceHive 360: built on SaaS, it runs multiple tenants within one structure plus uses custom AI modules made for specific tasks. Providers gain tools to list, adjust, and run offerings smoothly. Meanwhile, users explore options, weigh them side by side, then reserve spots using one clean portal. One piece of the setup involves a Workspace Concierge Agent - this one take care of bookings through everyday language plus works out prices smoothly. Instead of just reacting, the Dynamic Pricing Agent pulls together info about demand, markets, and how full places are to suggest smarter rates. Outreach happens automatically thanks to a combo system where lead generation meets sales activity, sending messages by SMS or email without manual steps. When it comes to suggestions, workers get workspace ideas suited to their travel patterns because another module pays attention to commute times and access routes. Feedback isn't ignored either; a Sentiment Analysis tool reads what users say so teams can act wisely later on. Behind everything sits ASP.NET Core, keeping things organized with separate but connected parts. Each AI function runs independently using FastAPI, letting them stay flexible and focused. Up front, Next.js shapes how people interact with the platform visually. Data splits neatly between two homes - one type fits into PostgreSQL like puzzle pieces while messier, freeform details go into MongoDB. Testing showed real gains: daily operations run smoother, visitors feel better supported, choices in business grow clearer as a result.

1. Introduction

One reason desks fill faster now? More people working solo, launching small businesses, or logging on from different cities choose shared offices instead of fixed rentals [1][2]. By 2023, those spaces brought in about $10.81 billion worldwide - estimates say that could jump to nearly $36.92 billion by 2030, growing just under 20% each year on average [3]. Numbers like these point one way: a shift toward open setups where teams mix and schedules stay loose.

Even with rapid expansion, trouble hides within the coworking world. Separate operations define most shared offices. One place runs bookings its own way. Another uses a completely different system. Data about members stays locked in isolated spots. Prices rarely shift to match what people want. Help arrives late when issues pop up. Without clear insight into trends, operators fly blind. Tools that forecast need? Missing altogether. Searching different sites to locate, weigh up then reserve spots means extra effort. Time drains fast, yet picks usually miss the mark [4][5].

Thinking machines come from computer science. These systems learn, decide, things people do. Changes are happening across fields right now. Coworking spaces use smart tools to handle work automatically. Leads appear through pattern spotting, not just luck. Room suggestions fit each person closely. Feelings in feedback get noticed by software. Prices shift when demand moves, moment by moment [6][7]. One system holding all this power reshapes how shared offices operate.

Out here, SpaceHive 360 shows up as an answer to real problems. Built on software you access online, it links different workspace operators into a single shared system. Smarts powered by artificial intelligence manage how prices shift, when people book, what users feel about services, also where they might go next based on movement patterns. While most tools stick to just one company's needs, this one open doors - letting space owners expand while visitors move smoothly from finding spots right through to reserving them.

This study dives into why SpaceHive 360 came to be, how it was built, rolled out, and tested. Following parts look at past research, suggest a structure through ideas on layout, walk through building steps, check outcomes, then point toward what might come next. Connected deeply, the shared workspace world and this tool shape one another. Problems in coworking spaces find answers here, through features meant to respond directly. Growth becomes possible when users get smoother interactions, shaped by real needs rather than guesses.

2. Literature Review

2.1. Growth and Evolution of Coworking Spaces

Out of simple beginnings, coworking areas now look more like organized hubs packed with tools people actually use. Not just freelancers show up anymore - entrepreneurs stop by too, along with remote staff and even company squads. These spots do more than offer desks they build connection through casual chats and teamwork that spark new ideas. People stay involved because they feel part of something real not just a room with chairs. What keeps users coming back isn't speedier Wi-Fi but the quiet sharing of tips and insights between neighbors who work nearby.

Out here, where cities grow fast, shared work spots matter a lot - especially for new businesses trying to get going [8]. Down in areas still building up, those spaces hand out office-like setups plus chances to connect, things tiny startups or solo workers might miss completely [9][10]. Over time, as the idea settles in, better software tools start feeling like a must-have just to keep everything running smoothly.

2.2. Artificial Intelligence at Work and Shared Offices

Workplaces today are shifting because artificial intelligence handles routine jobs, while also guiding quicker choices and shaping tailored user interactions [11]. Inside coworking environments, experimentation thrives - scheduling meetings now adapts through smart systems, real-time notes capture conversations, room settings adjust on their own, resource sharing grows more efficient [12].

Technology by itself falls short. Backed again and again by studies, human reactions hinge on how artificial intelligence acts in daily interactions [13]. Systems must own up to their role, shift smoothly when needed, also speak clearly without jargon. Just as vital - privacy. Workers notice which details get pulled in, plus ask reasons behind collection

[14]. Inside offices, evidence shows receptiveness exists, though strictly if tools simplify tasks instead of piling confusion atop routine [15].

Still, little focus has landed on AI behavior within shared office networks using several service suppliers. Research so far sticks close to one location or brand, skipping the tangled truth of markets where many vendors overlap [16]. Right there - in that cluttered zone - SpaceHive 360 takes its stance.

2.3. Multi-Tenant SaaS Architecture

Sharing one system among many groups defines SaaS, where each group stays isolated despite using identical tech. Their experience mimics a private installation - custom settings, unique processes - all untouched by others' activity. Behind it all, the operator runs the show efficiently, distributing expenses over every user [17]. One platform serves many, yet each acts alone. Putting together something like this takes work. Still, experts point to several key traits every solid shared platform must include. Each customer's actions should only touch their own information and configurations. Flexibility matters - every group using it ought to shape look and workflow themselves. When traffic grows or shrinks, the system adjusts on its feet. Even when packed, it runs. One user's details never cross into another's space. Features can differ per user, handled live, without breaking anything else [17].

Here's what gives all this real business impact: when a service shapes itself closely around individual user demands, it ends up supporting more users without stretching too thin - profit follows naturally [14]. That's the reason choices about structure during the first stages of building SaaS products often echo far into the future. One step ahead, newer ideas here lean into microservices setups - each tenant gets custom tools via separate systems while leaving the central platform untouched [15]. Instead of bulky single structures, this way manages growth and isolates breakdowns more cleanly; even smoother when AI tasks must operate apart from primary functions.

2.4. Estimating Work in SaaS Building

Getting a multi-tenant SaaS system built means thinking ahead about how much work it will take. Instead of guessing, one approach uses COSMIC sizing combined with UWE diagrams to predict workload early - before coding starts. This idea comes from research done by Ouh and Gan in 2023[17]. Their method was tested when turning a single-user app into a shared cloud service running on AWS. The numbers predicted lined up close to what actually happened during building. Measured error came out at just 0.11, which is below the usual pass mark of 0.25. When tenant needs keep shifting, being able to forecast time helps avoid surprises later. Even small changes per customer add up fast across many users.

2.5. Dynamic Pricing Meets AI Decisions

Most people know how flight costs shift each day. Prices rise when flights fill up fast. Hotels tweak room rates too. Ride services adjust during busy hours. These changes happen minute by minute. Coworking spaces mostly stay fixed though. Someone sets a price, then leaves it be. A closer look shows machines handle shifting numbers better than humans do. One person cannot track dozens of trends at once. Algorithms notice patterns across supply, need, rival moves, and timing. Smarter choices come from constant scanning. That kind of system already lifts income where used [18].

Some smart tools now adjust prices by studying customer interest, what competitors do, and how full spaces are - they aim to earn good income without losing their edge [16]. Because these systems run on their own but still avoid going below a minimum rate, they help keep earnings stable. This kind of setup turns out useful when running shared work areas. Profit safety stays intact even as choices get made faster. Not every digital helper does this well, yet the ones that follow such rules tend to last longer in real use. Decisions shift based on live inputs instead of guesses. That makes daily operations smoother over time.

2.6. Problems With Today's Shared Work Spaces

A big name in shared workspaces, Regus runs offices worldwide where people can rent desks or rooms by the day, week, or longer. With locations in many cities, its spaces come fully set up and move-in ready. Though widely available, access depends on being there in person - no virtual hub ties everything together. Other operators cannot join a central system to list offerings or handle

bookings through one service. Smart tools like learning algorithms, mood-responsive help, price shifts based on demand, or automated workflows aren't part of what they provide. Instead, operations stay grounded in physical sites without digital integration across providers. One roof for all workspace brands? Not here. Tech-driven adjustments to usage patterns or customer feedback do not shape the experience. Flexibility exists - but only within walls you have to walk into. Digital layers that connect supply, user behavior, and real-time decisions are missing entirely.

Most software made for shared workspaces handles reservations, member sign-ups, payment tracking, and how space gets assigned [19]. Though helpful when running one location alone, many of these systems stay separate - refusing to link different workspace owners together while missing smart features like adjusting prices based on demand or predicting future usage.

What's missing between these two systems? A smart system that pulls together many providers - exactly the space SpaceHive 360 steps into. Instead of staying empty, it brings intelligence where none existed before.

3. Approach and Suggested Framework

Inside SpaceHive 360, several workspace suppliers join one digital environment - each running their offerings independently while sharing the same smart backbone. A single front door lets users browse available spots, weigh options side by side, then secure space without switching screens. Intelligence weaves through every function here; it shapes rates based on demand shifts, adjusts availability in real time, learns from user reactions. Suggestions adapt quietly behind the scenes, shaped by past choices and current needs. This isn't bolted-on tech - it runs beneath everything, guiding decisions before they're made.

Built on separate modules, the system splits the main SaaS engine from its AI components. One piece at a time, each AI function grows on its own, then connects through a FastAPI layer. From there, the central backend pulls features using standard web requests. Because of this setup, updates to AI tools happen without touching the heart of the application. Independence means testing and fixing stay isolated, far from daily operations.

3.1. Proposed System Architecture

Out in the open, SpaceHive 360 builds its backbone using Clean Architecture - neat layers keeping things apart. Instead of stacking everything into one big block, which grows clumsy with many users and clunky when adding smart features, it spreads tasks across separate pieces.

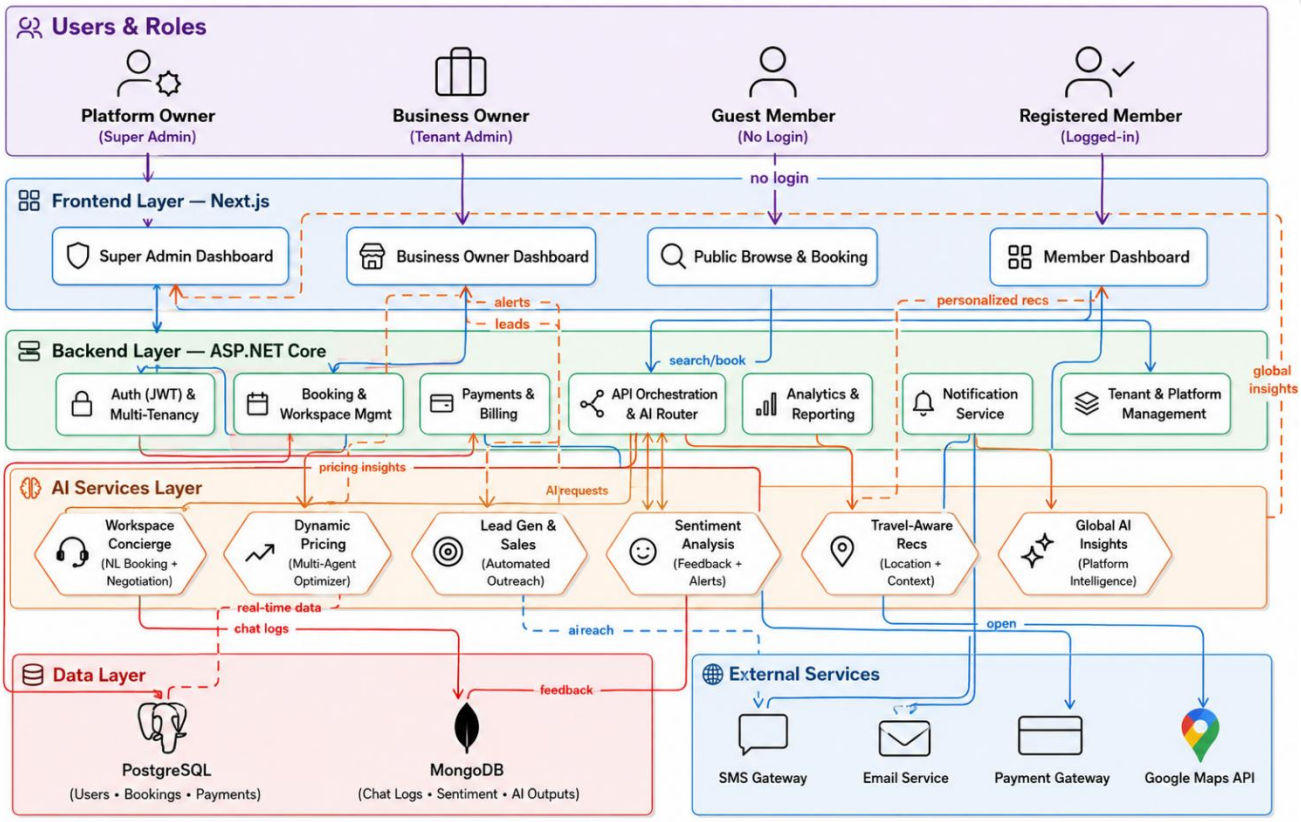


Figure 1: Proposed System Architecture

These parts chat via REST APIs, just loose enough to stay flexible. Going full microservices? That path meant too much operational noise for what this needs. So here it stands: balanced, connected, yet not tangled.

The system consists of the following layers as shown in Figure 1:

i. Frontend Layer: Built with Next.js, providing role-specific dashboards for Super Admin, Admin/Provider, and Member users. Real-time updates for bookings and notifications are handled via SignalR/WebSockets.

ii. Backend Layer: Implemented using ASP.NET Core (.NET 8), managing authentication (JWT-based), role-based access control, bookings, memberships, payments, and multi-tenant business logic. The backend is the central coordinator that routes requests to the appropriate AI services as needed.

iii. AI Services Layer: A set of independent Python projects, each handling a specific AI capability and exposed through FastAPI endpoints. The backend calls these services over HTTP when their functionality is required. Each AI project maintains its own internal logic, data access, and model operations without being embedded into the backend codebase.

iv. Data Layer: PostgreSQL is used for all structured, relational data including user accounts, workspace listings, bookings, memberships, provider configurations, and tenant data. MongoDB is used for unstructured data such as chat conversation histories, agent logs, and sentiment analysis records.

v. Workflow and Integration Layer: Automated workflows handle multi-step processes such as booking confirmations, lead outreach sequences, and escalation alerts. External integrations include payment gateways, SMS providers, and email services.

vi. Cloud and Deployment Layer: The system is containerized using Docker, designed for cloud deployment on AWS, Azure, or GCP, with CI/CD pipelines managing automated testing and deployment.

3.2. Multi-Tenant Design and Role-Based Access Control

One step inside, each coworking business lives in its own digital space. Not mixed up - kept apart by design. Through every layer, separation stays solid. Database rows wear invisible tags, seen only by their owner. PostgreSQL checks who's asking before showing anything. Behind the scenes, requests pass through filters tuned to one tenant at a time. What belongs to one never leaks into another. Settings shaped by AI follow the same rule. Bookings, memberships, workspace details - all walled off quietly. Even when systems talk, they speak only within bounds. Isolation runs deep, not just on top.

Access on the platform depends on one of three main user types. Who runs everything? That would be the Super Admin - overseeing every business account, handling subscriptions, watching analytics, adjusting system controls. Running a single workspace falls to the Admin or Business Owner - they set up areas, decide prices, organize team members, deal with client bookings and money matters, plus pull data-powered reports when needed. Then there are Members or Customers - these users explore available workspaces from different providers, reserve spots, handle their membership details, chat with the Workspace Concierge, and get suggestions tailored just for them.

3.3. AI Systems

Inside SpaceHive 360, artificial intelligence runs through five separate systems. Each one began life as its own Python build. FastAPI opens them up to outside connections. Taken as a whole, these parts manage every phase of running a shared workspace. Bringing in fresh users happens here. So does keeping those already on board. Pricing decisions form part of it too. Then there is grasping what people really think about how things feel when they use the space.

3.3.1 Workspace Assistant and Deal Coordination Setup

A chat-based tool called the Workspace Concierge Agent lets people look for and reserve office spaces by typing what they want in everyday words [20]. Rather than complete structured inputs, someone might say they need a quiet spot downtown for four workers under $100 daily. Once it grasps the details - like space kind, headcount, area preference, or price range - it scans available places through meaning-aware queries. Then, matching results pop up based on how closely they fit the described situation.

Most useful thing here? It handles talks between people. Once someone picks a spot, chatting about cost begins right there. Everything said stays in view, tied just to that place chosen. Instead of a real person, a built-in helper reviews what's offered. Response comes fast - could be yes, no, or a new number back. Back-and-forth rolls like this till things match up or one walks away.

Now here's how it picks up what matters - things like your price range, group size, spot on the map, plus if you're looking to haggle or just reserve a slot. Even when you say it differently than the database expects, meaning-based search still pulls up what fits. Though phrasing changes, matches stay accurate.

3.3.2 Dynamic Pricing Agent

Starting with smart adjustments, the Dynamic Pricing Agent works like a team of helpers figuring out ideal prices for shared work areas. Because it watches how busy places get, what competitors do, and current renting trends, its suggestions shift over time. One moment it responds to high traffic, another moment it reacts to quiet periods. Guided by real-world data from Luo and others in 2024, the system stays grounded in actual patterns. Instead of fixed numbers, pricing flows more naturally through constant small updates.

Once switched on for a location, its past reservations get examined while three separate agents start working at once

- Demand Agent: Studies past bookings to find demand trends and peak usage times. It also uses previous analysis stored in memory.
- Market Agent: Checks external signals like competitor prices and market trends in cities such as Karachi or Lahore.
- Occupancy Agent: Looks at current occupancy levels and available space to understand utilization and risk.

Memory keeps every outcome inside each workspace, shaping sharper choices later. Running

via Groq API, a big language engine shapes the responses into clear formats.

3.3.3 Sentiment Analysis Module

Checking how users feel comes from looking at what they write - reviews, chats, messages, or notes when booking. Their words give clues about mood through a special system that reads between lines. A label of positive, neutral, or negative gets assigned to every piece of text - alongside that, the core cause driving the emotion shows up clearly. What stands behind the feeling comes into view just after the tone is set. When complaints pile up, a signal flashes on the admin screen. That way problems get sorted fast - before they spiral. Later on, it tracks how feelings shift across months, giving businesses a clearer picture of what users really think. Still, patterns emerge that reveal deeper reactions than just single comments show alone.

3.4.4 Travel-Aware Recommendation System

Not only does it look at how far places are, but location habits shape its choices too. What matters most isn't mileage - it's movement patterns over time. Suggestions shift when routines change, not just because of proximity. Instead of fixed rules, behavior guides what shows up. Where you go often plays a bigger role than where is nearest.

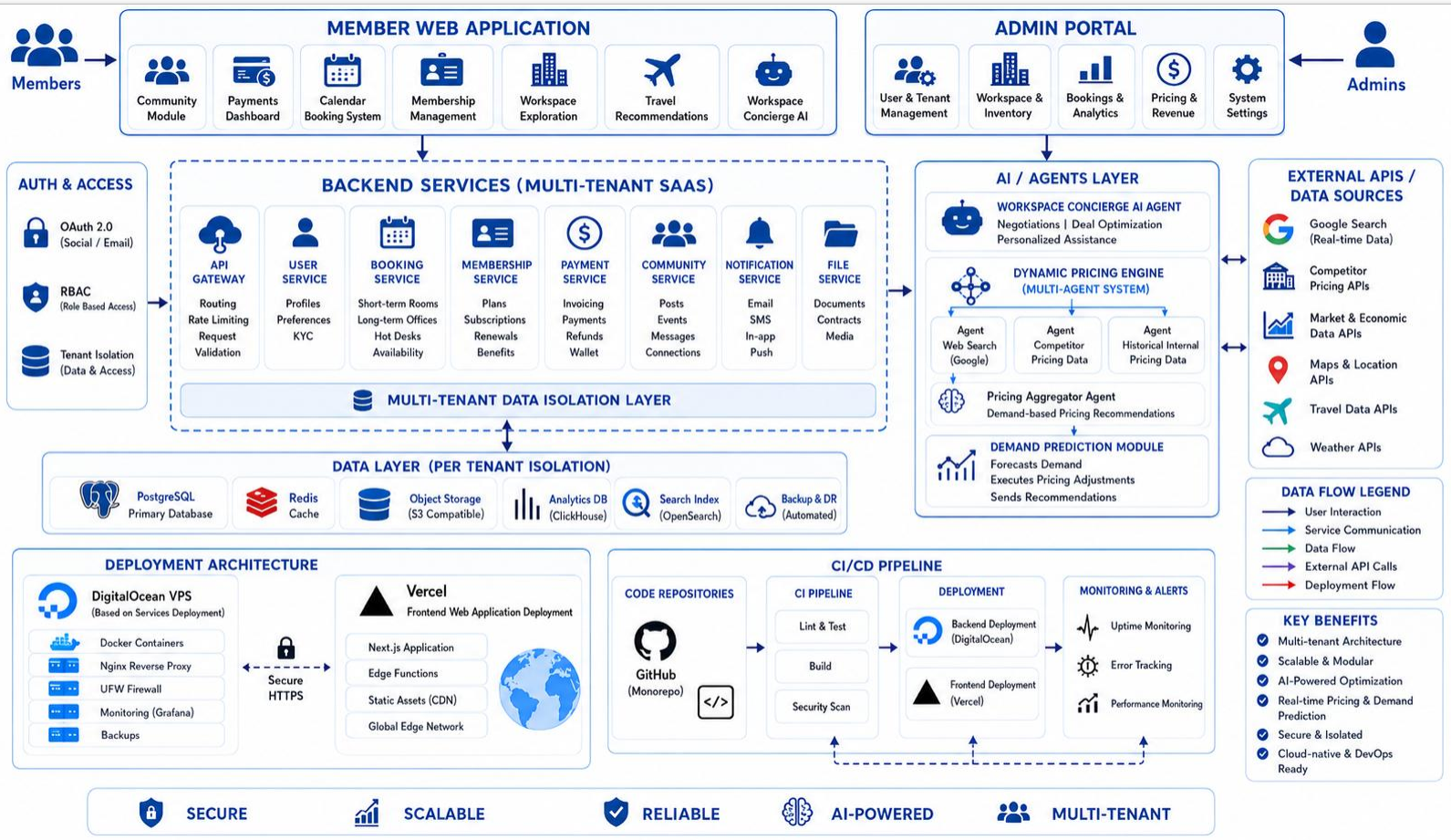
It looks at the following factors i. travel time, ii transport cost, iii.mode of transport and iv. user needs and purpose. A space farther away might still appear first if getting there is quicker. Convenience shapes the order, not just distance. A location with better transit links could beat one that's nearer but tricky to reach. Users get advice that fits better with real life. Not every tip works for everyone, yet most suggestions here match daily routines well.

4. Implementation Methodology

Figure 2 shows the implementation methodology.

4.1. Development Process

Working on the project meant starting fast with small steps, yet keeping clear plans along the way. Some parts moved quickly through trial rounds while still holding onto organized write-ups. Progress came from short loops of testing followed by steady record-keeping.



Flexibility mattered but so did having things written down properly. Every few weeks, tasks got packed into tight cycles. Each cycle produced live pieces - like databases, API links, AI workflows, or screen layouts. The supervisor checked progress often, sitting in on these short demos. Their comments shaped what came next, slipping into future versions without delay. One person handled planning, while another took charge of communication; the third managed tasks on site. Backend development and project coordination using ASP NET Core with databases and APIs. Whereas Frontend and UI UX with Next js dashboards Tailwind CSS SignalR and AI automation is implemented with with Python FastAPI AI agents NLP systems. Figure 4 shows application front end

4.2 Backend Implementation

Built on ASP.NET Core with .NET 8, the backend follows a clean architecture pattern. Layered into domain, application, infrastructure, and API sections, it organizes code by purpose. Business rules stay isolated here, making updates smoother. Testing becomes simpler since components are split by function. Each part connects without depending too tightly on others.

Tokens made of JWT handle who gets in, each person tagged by role while tenants stay apart through smart routing layers. Booking fees plus membership costs move via Sandbox lanes built for testing money flow. From the back end, REST APIs feed into both the Next.js front end along with AI tools. Updates that happen live - such as changes in booking state or alerts - run through SignalR via WebSockets.

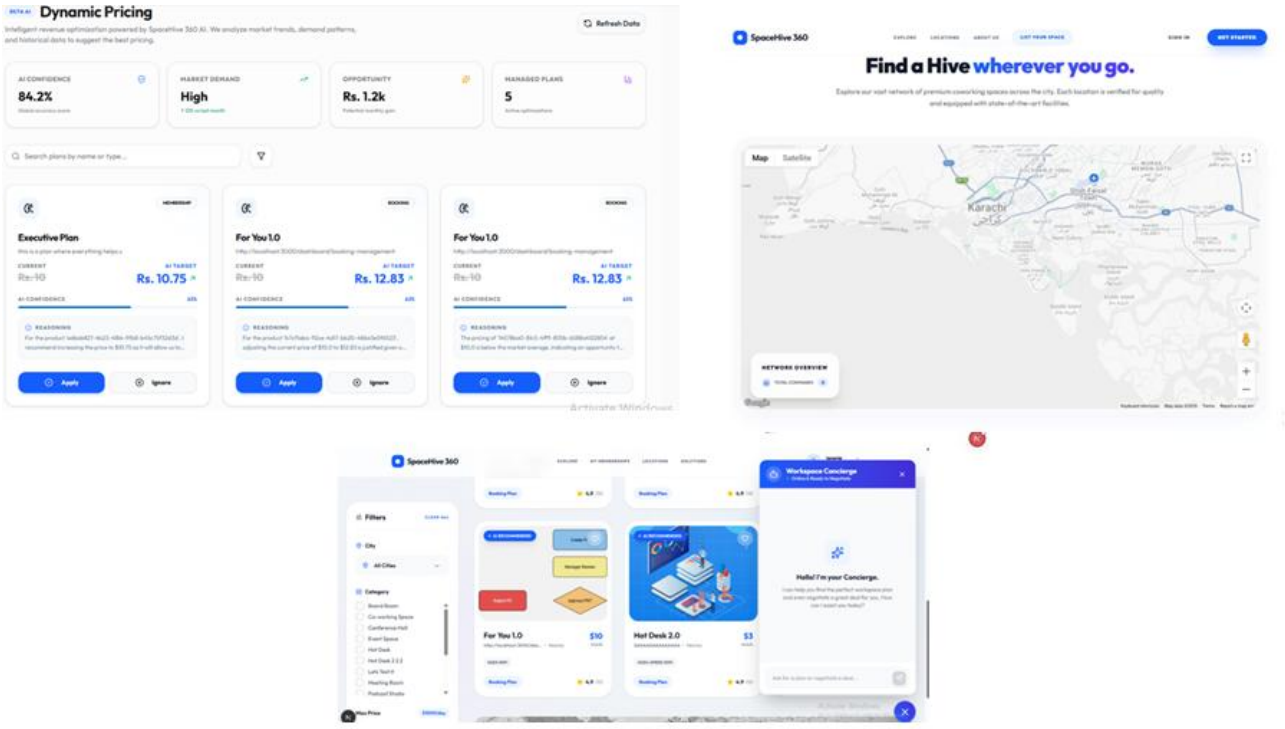


Figure 3: Application front end

4.3 AI Services Implementation

A single AI setup lives inside its own Python FastAPI container, built apart from others on purpose. One piece works without needing another, shaped separately each time. A smart helper for your work space starts by reading what you need through meaning-aware searches. It uses Chroma DB to dig into requests deeply instead of just matching keywords. Understanding intent comes next, shaping how it responds to questions or tasks. Booking rooms or discussing terms unfolds smoothly because it knows context matters. Requests shift from vague asks to clear actions thanks to focused interpretation. This system links needs with answers without relying on exact phrases. Smooth coordination happens behind the scenes when intentions are truly grasped.

Pricing shifts happen through separate AI pieces - each one watches demand, market trends, or room fill levels. One after another, they feed findings into a central decision module. This main part weighs what each has found, then shapes a price suggestion. The system leans on Llama 3.3 70B, accessed using Groq's API. Outputs come back locked into clean JSON format, built for precision. Feedback gets checked by a tool that reads emotions. This system uses smart reading tricks to guess if words feel good, okay, or bad. Results go into a storage spot called MongoDB. Dashboards pull these feelings later. Each piece waits ready to show up when needed.

When you move around, it picks spots based on how long it takes to get there. Getting there by bus or train changes what shows up first. Price matters just as much as distance when sorting places. It checks real-time routes through Google Maps to decide rankings. Time spent commuting shapes which workspace feels most convenient.

4.4 Data Layer

The system uses a hybrid database approach:

Structured information like user profiles, reservations, financial records, and space configurations live inside PostgreSQL. Row-based safeguards make sure each tenant stays separated. What keeps data organized is its strict format rules. Security at the record level prevents cross-tenant

access. Information remains grouped by client without mixing. Isolation happens directly within how rows are handled.

Out here, MongoDB handles messy info - think chat records, emotional tones, or what AI agents spit out. It bends instead of breaking when data changes shape. Stuff piles up without needing a fixed plan. Loose structure means it keeps going even when inputs get weird. Flexibility comes from how it skips rigid rows. Data lives freeform, fitting shifts on the fly. No templates required, just raw pieces slotting in as they arrive.

4.5 Frontend Implementation

Frontend runs on Next dot js alongside Tailwind for styling. One dashboard serves admins, another handles user views three categories are i. Super Admin, ii,Workspace Admin/Provider and iii.Members/Customers. Booking stats show up on dashboards alongside smart suggestions powered by artificial intelligence. Alerts pop up when customer mood shifts, tracked through feedback signals. Leads move step by step with clear visibility at every stage. Search inside workspaces knows about trips, adjusts results based on location needs. Updates happen instantly thanks to SignalR pushing data live. Visuals take shape using Chart.js along with Recharts building dynamic displays.

5. Results and Evaluation

5.1 Overview

Testing SpaceHive 360 focused on three main parts: the software platform, its five AI components, then how everything works together. Because development is still wrapping up, findings show how individual modules behave, whether AI responses make sense, alongside performance during tests that mimic real-world usage with multiple users at once. Checking it fully using actual user traffic will happen later.

5.2 Core SaaS Platform

Authentication checks came first when testing the backend running on ASP.NET Core (.NET 8). Role-based permissions followed during evaluation of core functions. Data separation across tenants mattered just as much in the process. Booking sequences were examined closely afterward. Payment handling rounded out the main tasks under review.

Through simulated operations of several provider accounts at once, Multi-Tenant Isolation got tested under pressure. While one tenant accessed their own information, row-level controls inside PostgreSQL blocked any view into others' spaces, reservations, or user details - no matter how the tenant ID in requests had been altered. Even with tampered inputs, zero data slipped across boundaries during checks. This outcome lines up precisely with what earlier studies on shared-service platforms demanded: strict separation of resources among users [17].

Super Admin, Admin/Provider, and Member - all went through Role-Based Access Control checks. Only their allowed areas opened up, nothing more. Providers touched just one tenant's details, never another's. The top-level admin saw everything across the system. When someone tried reaching locked endpoints, walls stayed solid. Responses shut them out cleanly, every single time.

Start to finish, the booking system handled searching for available spots, making reservations, sending confirmations, then cancellations - all without hiccups during test runs. Money transactions went through fake payment portals just fine, while live updates popped up on user screens instantly thanks to SignalR pushing fresh statuses the moment they changed.

5.3 AI Systems Evaluation

5.3.1 Workspace Concierge Agent

Sometimes folks asked in roundabout ways, yet the agent still found what mattered - room kind, space needed, where, how much. No matter if questions felt loose or tight, wording shifted wildly but results stayed strong. Even when people used casual talk, the details popped out clear each time. Phrases changed shape constantly, still it pulled room type, headcount, place, price without fail. Most attempts worked fine, though wordings came in all forms, vague or sharp alike. Details emerged right nearly every single go, regardless of how things were said. Through messy speech patterns, the core needs surfaced just like that. Different styles arrived daily, yet extraction never missed a beat on those four points. Whether short or stretched out, the answers held steady in outcome. Every odd way of asking still led to correct pieces being picked up properly.

From time to time, test talks followed a set script through several back-and-forths. Through these exchanges, the system held on to the topic at hand, stayed within the chosen project area, while shaping replies that made sense. When someone tried offering too low a figure, it responded by saying no - just as designed. Prices settled upon moved smoothly into the next phase - the reservation stage - with nothing lost along the way.

Even if what people typed didn't line up word for word, the system still pulled correct workspaces most of the time. Its grasp of meaning showed through during testing, pulling matches based on intent rather than just keywords.

5.3.2 Dynamic Pricing Agent

A fresh look at the Dynamic Pricing Agent came through tests using made-up booking patterns. Not just one, but several trial spaces faced changing crowd levels over time. Instead of real data, computer-driven situations helped shape each round. Each workspace saw different use peaks and quiet spells. Through these runs, pricing shifts were tracked closely. What emerged depended on how full or empty spots became. Simulated stays shaped every price move seen.

One after another, the trio - Demand, Market, Occupancy - delivered clean results quickly, powered by Llama 3.3 70B through Groq's interface. Out of those pieces came a single suggestion, shaped by reasoning and tagged with certainty, pulled together not randomly but on purpose by the Strategy Agent. Every situation stayed within bounds; never once did any outcome dip under the set price limit, something absolutely required when putting systems like this into live use.

Later checks confirmed memory stayed intact through multiple uses. The setup used earlier findings each time it ran. With added details over sessions, advice got slightly better. Past work shaped what came next without needing repeats.

5.3.3 Sentiment Analysis Module

A handful of user replies - pulled from pretend reviews, booking notes, and chat logs - were sorted by hand to check how well the Sentiment Analysis Module performed.

Most samples got sorted right - into positive, neutral, or negative buckets. When harsh comments came through, the tool spotted negativity fast enough to notify admins without delay. Tough messages landed where they should - the high-priority section of the admin view. Every emotion tag made it into the database just as planned, shaped properly for later graphing.

5.3.4 Travel-Aware Recommendation System

A handful of travelers helped test the system, their habits and spots spread wide. From somewhere new each time, arrival lengths and price tags came through clear when tied to Google Maps. Details stayed sharp, pulled straight from the road ahead.

A spot farther out often showed up higher in results when it simply took less time to reach, even if nearer places existed. Getting there was cheaper too sometimes, which helped its position despite the extra miles. What mattered most became clear through these tests - ease of access beat short distances every time. The goal all along had been about usefulness, not just what looks close on a screen. Real movement patterns shaped the outcome more than straight lines between points ever could.

5.4 System Integration

From start to finish, tests checked how the ASP.NET Core backend talked with each of the five FastAPI AI tools using REST. Well-shaped JSON came back every time an AI service answered, letting the backend understand and respond properly. When tests followed a clear pattern, everything worked without a single breakdown.

Surprisingly smooth, the hybrid data layer kept up. PostgreSQL took care of structured transactions without a hitch. Meanwhile, chat logs, sentiment records, and agent results landed safely in MongoDB. Schema issues never showed up. Unstructured AI output fit right in.

5.5 Results Comparison With Other Platforms

Looking at how things stand now, today's coworking tools fall short in smart workspace handling when lined up against alternatives. Table 1 shows comparison with other platforms. While some platforms handle reservations and daily tasks well, others dive into AI possibilities within shared offices - mostly seen in research settings. Commercial options often stick to logistics, whereas university studies tend to probe deeper into machine learning uses for office spaces. Even

though cloud-based setups serving multiple users are common in tech discussions, they rarely bring together self-guided decisions powered by AI or services that run without constant human input.

What stands out when looking elsewhere? Missing pieces pile up. A shared space booking hub does not exist nearly anywhere. Chat support powered by smart systems feels like an afterthought, if present at all. Prices stay fixed, no room to discuss terms. Reading emotions in user feedback barely happens. Generating interest and reaching out runs on manual effort mostly. One thing becomes clear through it all - stitching every piece into one thinking ecosystem hasn't been done yet.

Facing those gaps, SpaceHive 360 steps in with a design where separate AI helpers work across shared spaces - suggesting spots, adjusting prices live, reading user moods, guiding guests, sparking new leads. Instead of standalone tools, it weaves together market features, smart workflows, and linked artificial intelligence into one connected system. Testing shows it handles growth better, covers more needs, runs smoother compared to what's out there now. Because everything talks to everything else, managing flexible offices becomes faster, sharper, less tangled. Not just another upgrade - it reshapes how these systems think, react, stay in sync. What used to lag behind demand now moves ahead of it.

Table 1: *Comparison Of Results*

Study / Platform	Domain	Multi-Tenant SaaS	AI Integration	Conversational Agent	Dynamic Pricing	Sentiment Analysis	Recommendation System	Lead Generation & Sales Automation	Marketplace (Multi-Provider)
Regus [21]	Coworking	✖	✖	✖	✖	✖	Basic Search	✖	✖
Spacebring [22]	Coworking SaaS	Partial	Limited	✖	✖	✖	Basic Search	✖	✖
Nexudus [23]	Coworking SaaS	✔	Limited Automation	✖	✖	✖	Basic Recommendations	✖	✖
Abrar et al. (2025)[16]	AI in Coworking	N/A	✔	Conceptual	✖	✖	Conceptual	✖	✖
Zirar et al. (2023)[24]	AI Workplace	N/A	✔	Partial	✖	✔	Partial	✖	✖
Berbegal-Mirabent (2021)[25]	Coworking Research	N/A	✖	✖	✖	✖	✖	✖	✖
Ouh & Gan (2023) [17]	Multi-Tenant SaaS	✔	✖	✖	✖	✖	✖	✖	N/A
Song et al. (2018) [26]	SaaS Architecture	✔	✖	✖	✖	✖	✖	✖	N/A
Fukumura et al. (2021) [27]	Workplace AI	N/A	✔	✖	✖	✔	✖	✖	✖
Bouncken et al. (2021) [28]	Coworking Management	N/A	✖	✖	✖	✖	✖	✖	✖

SpaceHive 360 (Proposed)	AI-Powered Coworking SaaS	✔	✔	✔	✔	✔	✔ Travel-Aware	✔	✔

5.6 Limitations Of Current Evaluation

Being clear about the limits of this test matters. Every check happened using made-up or example data, not actual large-scale usage by people (Abrar et al., 2025). These AI systems haven't faced the messy flow of true operational input yet, so how well they perform outside the lab could be different. When many users hit the system at once - thousands across several groups - the behavior stays unknown because those conditions were never measured. That missing piece? It's something later efforts must tackle.

6. Discussion

6.1. Filling the gap between research and market

So far, studies about AI in shared workspaces have mostly overlooked hybrid setups involving several service providers - until now. What makes SpaceHive 360 stand out is its ability to run distinct AI tools across different platforms, not only inside one location. Instead of isolated functions, it weaves together smart pricing, automated outreach, mood tracking, chat-based reservations, and tailored suggestions through a single software hub. This setup pushes beyond what competing offerings deliver today. While others stick to basic features, this approach links capabilities in ways rarely seen before. Behind it all lies a system built for flexibility, connecting dots where past solutions fell short.

6.2. AI at the center of operations not added on

What stands out in SpaceHive 360's layout is how AI runs beneath everything, not added on top. Since every AI piece fits one clear purpose, it uses live data from the system, feeding results straight into choices made by staff or software. Because integration matters, studies show tools like this work better inside daily routines instead of standing apart. Seamless blends beat isolated features, according to Abrar and team in 2025. Behind the scenes, function shapes form - each process shaped by constant inputs, never working blind. Most tools just sit there doing nothing until someone decides to act. Not this one. Priced adjustments happen by default each cycle without needing permission first. Memory keeps earlier outcomes so changes build on what worked before. Decisions evolve because the process repeats under normal operations. Results feel helpful since thinking ahead is baked into daily functions already. Help arrives even when nobody asks for it.

6.3. The SaaS and AI Balance

SpaceHive 360 puts AI front and center, yet its backbone rests just as much on standard software features. Because behind smart algorithms sits a structure - multi-user setup, permissions set by roles, tracking spaces, handling bookings, managing memberships, moving payments. These pieces feed information so anything intelligent has something real to work with. When the base stumbles - say messy details, shaky reservation logs, missing room facts - the smarts start slipping too. What comes out depends entirely on what went in first. Without solid ground below, even sharp models falter.

Because everything connects, getting the SaaS base right matters more. Start with proven design rules - keeping tenants separate, scaling smoothly under load, handling shifting demands (Ouh & Gan, 2023). Only after that fits, add AI. Shape grows from structure.

6.4. Ethical Considerations

Out there among digital footprints, SpaceHive 360 picks up pieces like how people behave, what they type, their emotional cues, alongside where they log in from. Because of this, treating data right isn't just expected - it shapes every decision. Separate walls keep each tenant's info apart by default, while PostgreSQL holds everything locked down tight. Built-in rules favor privacy first, not as an afterthought but at the core. When someone talks to the Workspace Concierge, it's machine intelligence on the other end - no pretending otherwise. Clear signs let users know they're chatting with software, which studies show helps trust grow naturally over time (Abrar et al., 2025).

Worries about privacy kept showing up alongside questions of how much machines should decide versus people in job settings, according to Zirar and team in 2023. At SpaceHive 360, artificial intelligence lends a hand by offering guidance instead of making moves on its own. Before any price change goes live, someone in charge checks it first. When the system spots tension in customer messages, it raises a signal - yet leaves the response to staff. People stay in control even as smart tools help things run smoother.

6.5. Limitations

Backed by a student's drive, SpaceHive 360 took shape during senior-year weeks when hours were tight. Real-life details stayed sparse because actual field numbers never fully lined up. Tools on hand shaped what could be built, since access to gear had clear boundaries. Time pressed hard - what emerged fits within those fixed walls.

Most times, testing happens with fake numbers because actual big-data examples aren't available yet. The setup isn't live, so trials run on made-up inputs rather than genuine records pulled from daily use. Without access to true volumes of active user activity, results might miss the mark when guessing real habits or company needs.

Heavy loads haven't pushed the system yet - no runs with thousands using it at once, or many tenants operating together in actual daily use. So how fast it runs, whether it scales well, or holds up when crowded stays partly unknown.

Right now, the AI pieces grow under tight oversight, so what they produce follows set rules or mimics earlier test runs. When used out in the world, such systems must keep learning - fed fresh information - to sharpen how well they judge and respond.

7. Conclusion

One thing becomes clear when looking at SpaceHive 360 - building a smart, shared software system for coworking spaces can actually work. Instead of isolated tools, several operators join a unified online environment where technology handles complex tasks behind the scenes. Five distinct artificial intelligence components power the setup: one chats with users to help book and haggle over desks, another adjusts prices on the fly using group decisions, while a third reads customer feelings from feedback. A fourth suggests spots based on where people are traveling from, yet each piece connects smoothly into daily operations. What used to be scattered apps now behaves like one thinking organism. Old problems such as clunky interfaces, guesswork pricing, lack of insight, and disconnected services fade once these layers come together. Though many attempts fail by adding only small fixes, this approach rebuilds the foundation. Intelligence runs through every part rather than sitting on top as an afterthought. Surprisingly, mixing live data with automated logic makes routine actions feel responsive. The gap between what users expect and what offices deliver shrinks without grand promises. Efficiency emerges quietly because coordination happens before anyone asks. After all, simplicity often hides inside well-linked parts working unseen.

Working nonstop on live data makes AI perform better here. Not when tucked away analyzing bits now and then. What holds the system up? A tight bond between the service base and the smart layer riding on top. Each step forward pulls the other along. Users get more out of it only if both sides keep improving at once.

With the coworking sector growing fast (Abrar et al., 2025), expectations rise for workspaces that are both intelligent and seamlessly linked. Instead of standing still, progress pushes toward tools like SpaceHive 360 - built to adapt, grow, and roll out widely. This approach doesn't just respond, it fits where needs shift quickly.

8. Future Work

One path forward involves upgrading SpaceHive 360 so it thinks faster, reacts instantly. With time, the system could learn on its feet, adjusting while things happen instead of after. Real environments bring chaos - future versions aim to handle that mess without breaking stride. Progress means building smarter layers, ones that mirror how problems unfold outside labs. Instead of waiting, responses become immediate, shaped by live shifts in data flow. Closer ties to actual conditions make outcomes less predictable but far more useful.

Most people find it easier to book a desk when they can do it from their phone while moving around. Instead of just logging in later, the system gives smart picks right away if it knows where someone stands, what traffic looks like nearby, or which

spots open up first. Location signals help fine-tune those alerts without needing extra steps. The moment shifts into action, guidance arrives through quiet updates tied to movement.

Later updates might bring IoT sensors into shared work areas. These gadgets could track how rooms are used, moment by moment. Temperature readings may flow alongside sound measurements. Instead of guessing when spots will fill up, systems learn from live signals. Room use patterns help shape better layouts over time. Machines adjust lighting or cooling based on what they sense. Noise trends might influence design choices down the line. Live feedback loops let buildings respond like living things. Energy tracking becomes part of daily operations. Smart features grow quietly behind the scenes

Cameras feed data into computer vision setups by watching how areas get used. Take seating levels, room activity counts, time-based crowding shifts - these snap into view instantly. Automation kicks in once the system logs who's where, no human checks needed. Decisions on cost settings adjust smarter when live use signals guide them. Room layouts reshape themselves behind the scenes, driven by actual foot traffic trends instead of guesswork.

9. References

[1] J. Brown, "Curating the 'Third Place'? Coworking and the mediation of creativity," *Geoforum*, vol. 82, pp. 112–126, 2017, doi: 10.1016/j.geoforum.2017.04.006.

[2] M. Mayerhoffer, "Growth factors of the coworking industry: the case of Prague," *Journal of Property Investment and Finance*, vol. 38, no. 3, pp. 203–212, 2020, doi: 10.1108/jpif-12-2019-0164.

[3] N. M. S. Consulting, "How is the co-working space market transforming the future of work," *Zenodo*, 2025, doi: 10.5281/zenodo.18092551.

[4] S. Ivaldi, A. Sannino, and G. Scaratti, "Is 'co' in coworking a short for contradictions?," *Qualitative Research in Organizations and Management: An International Journal*, vol. 17, no. 5, pp. 38–63, 2021, doi: 10.1108/qrom-06-2020-1970.

[5] S. U. Naima *et al.*, "Literature study: Increasing productivity in co-working space," *Indonesian Journal of Visual Culture Design and Cinema*, vol. 4, no. 1, pp. 32–38, 2025, doi: 10.21512/ijvcdc.v4i1.12962.

[6] R. Dash *et al.*, "Application of artificial intelligence in automation of supply chain management," *Journal of Strategic Innovation and Sustainability*, vol. 14, no. 3, 2019, doi: 10.33423/jsis.v14i3.2105.

[7] S. Mathur *et al.*, "Automating the future: user insights on smart coworking spaces," in *Proc. IEEE IC3ECSBHI*, 2025, pp. 502–506, doi: 10.1109/ic3ecsbhi63591.2025.10991018.

[8] P. Bednář and L. Danko, "Coworking spaces as a driver of the post-Fordist city: a tool for building a creative ecosystem," *European Spatial Research and Policy*, vol. 27, no. 1, pp. 105–125, 2020, doi: 10.18778/1231-1952.27.1.05.

[9] A. Füzi, "Co-working spaces for promoting entrepreneurship in sparse regions: the case of South Wales," *Regional Studies, Regional Science*, vol. 2, no. 1, pp. 462–469, 2015, doi: 10.1080/21681376.2015.1072053.

[10] S. L. Gall *et al.*, "The coworking space: a catalyst for initiatives at the crossroad of mobility and embeddedness. Lessons from peripheral areas of western France," in *Edward Elgar eBooks*, pp. 226–243, 2024, doi: 10.4337/9781802209181.00027.

[11] M. Bisht, "Transforming the modern workplace: the impact of artificial intelligence on office work efficiency and employee dynamics," *International Journal of Engineering Development and Research*, vol. 13, no. 3, 2025, doi: 10.56975/ijedr.v13i3.301566.

[12] A. Varol *et al.*, "Creation of AI-driven smart spaces for enhanced indoor environments – a survey," *Internet of Things*, vol. 36, p. 101876, 2026, doi: 10.1016/j.iot.2026.101876.

[13] M. Mirbabaie *et al.*, "The rise of artificial intelligence – understanding the AI identity threat at the workplace," *Electronic Markets*, vol. 32, no. 1, pp. 73–99, 2021, doi: 10.1007/s12525-021-00496-x.

[14] S. Corvite *et al.*, "Data subjects' perspectives on emotion artificial intelligence use in the workplace: a relational ethics lens," *Proc. ACM Hum.-Comput. Interact.*, vol. 7, pp. 1–38, 2023, doi: 10.1145/3579600.

[15] O. Hornung and S. Smolnik, "AI invading the workplace: negative emotions towards the organizational use of personal virtual assistants," *Electronic Markets*, vol. 32, no. 1, pp. 123–138, 2021, doi: 10.1007/s12525-021-00493-0.

[16] M. Abrar *et al.*, "The evolution of co-working spaces with artificial intelligence," *SSRN Electronic Journal*, 2025, doi: 10.2139/ssrn.5239699.

[17] E. L. Ouh and B. K. S. Gan, "An exploratory study of architectural style and effort estimation for multi-tenant microservices-based SaaS," in *Proc. IEEE ICSA-C*, pp. 159–166, 2023, doi: 10.1109/icsa-c57050.2023.00043.

[18] A. Zirar, S. I. Ali, and N. Islam, "Worker and workplace AI coexistence: emerging themes and research agenda," *Technovation*, vol. 124, p. 102747, 2023, doi: 10.1016/j.technovation.2023.102747.

[19] L. Remes *et al.*, "Smart workplace solutions – can they deliver the offices that employees have been waiting for?," *Facilities*, vol. 40, pp. 40–53, 2021, doi: 10.1108/f-04-2021-0032.

[20] P. Priya *et al.*, "GENTEEL-NEGOTIATOR: LLM-enhanced mixture-of-expert-based reinforcement learning approach for polite negotiation dialogue," in *Proc. AAAI Conf. Artificial Intelligence*, vol. 39, no. 23, pp. 25010–25018, 2025, doi: 10.1609/aaai.v39i23.34685.

[21] Regus, "Flexible office space and coworking solutions," 2025.

[22] Spacebring, "Coworking management software," 2025.

[23] Nexudus, "Coworking space management platform," 2025.

[24] A. Zirar, S. I. Ali, and N. Islam, "Worker and workplace AI coexistence: emerging themes and research agenda," *Technovation*, vol. 124, p. 102747, 2023.

[25] J. Berbegal-Mirabent, "What do we know about co-working spaces? Trends and challenges ahead," *Sustainability*, vol. 13, no. 3, p. 1416, 2021.

[26] H. Song, F. Chauvel, and A. Solberg, "Deep customization of multi-tenant SaaS using intrusive microservices," in *Proc. ICSE*, 2018.

[27] Y. E. Fukumura *et al.*, "Worker perspectives on incorporating AI into office workspaces," *Int. J. Environ. Res. Public Health*, vol. 18, no. 4, p. 1690, 2021.

[28] R. B. Bouncken, M. M. Aslam, and Y. Qiu, "Coworking spaces: understanding, using, and managing sociomateriality," *Business Horizons*, vol. 64, no. 1, pp. 119–130, 2021.