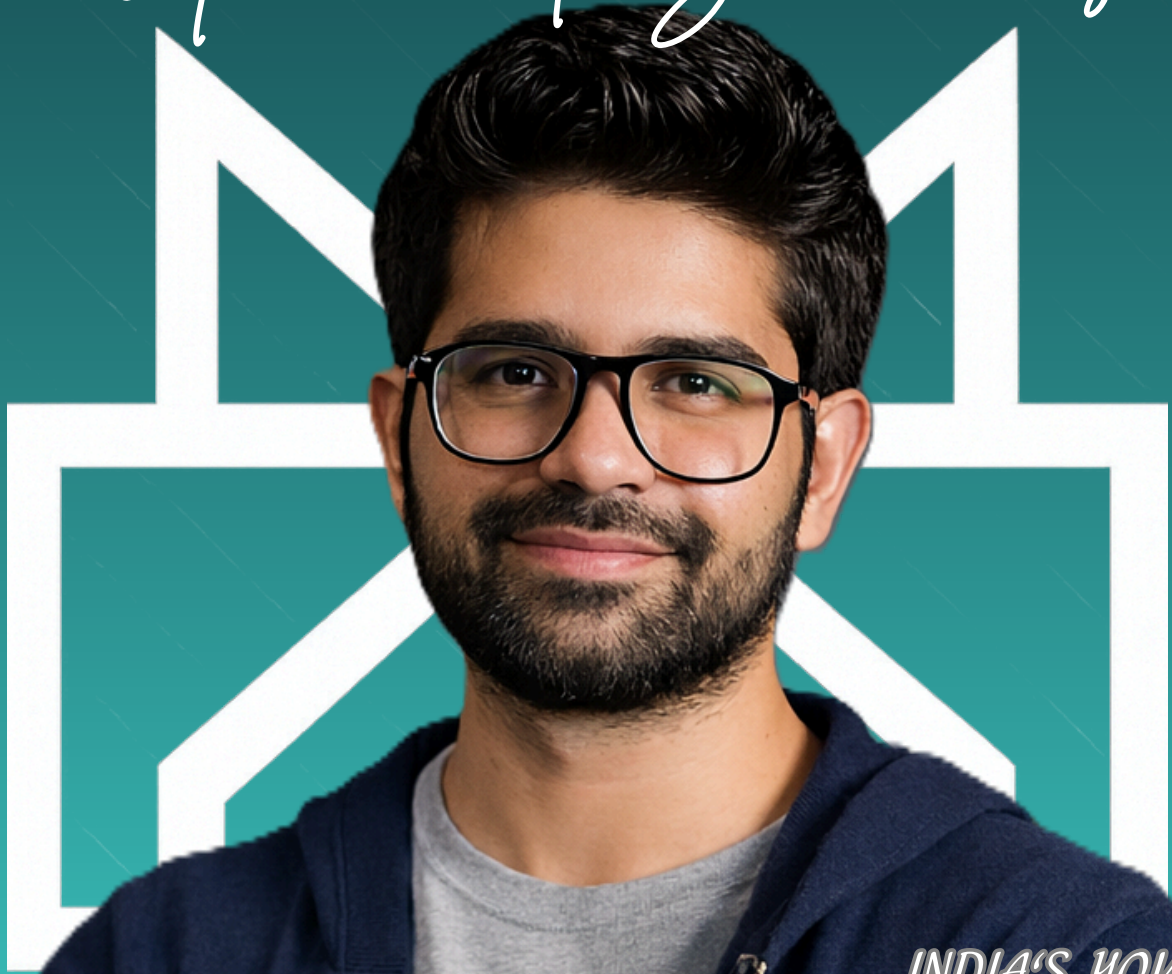




Vishwakarma Government  
Engineering College  
Department of Information Technology

Aravind Srinivas



PERPLEXITY CEO

INDIA'S YOUNGEST  
BILLIONAIRE

THE

TECH TREASURE

VOLUME 06 '25

# VISION

---

The department aims to nurture students to be IT professionals who are innovative and technically competent to meet the advancements in industrial and societal needs.

# MISSION

---

To produce technically competent and socially responsible graduates.

To develop well-equipped laboratories to gain practical exposure in IT practices.

To support research culture, consultancy work and enhancement of entrepreneurship skills.

To enhance industry linkage for good placements and real-world exposure.

To encourage, involve and support students for environmental and social wellbeing.

# PEOs

---

The graduate will be a competent professional with high-level technical proficiency in the field of Information Technology to identify problems and develop innovative solutions to meet the industrial needs.

The graduate will exhibit professionalism, teamwork, leadership skills, and lifelong learning .

The graduates will ethically apply their computing knowledge and skills considering societal, economic, and environmental factors.

The graduate will be intellectually competent to pursue higher education and conduct research-oriented activities.

# PSOs

---

Demonstrate proficiency in one of the emerging Technology like Data Science, Machine Learning, and Web/Mobile Application Development.

Demonstrate skills to design, develop and test software systems to provide solutions to real-world problems.

# “Message from HOD”

## Welcome to Tech Treasure 6.0

Information Technology is no longer just a tool we use, it is the foundation on which we build our future. Today, we stand at a time where technology connects intelligence with humanity, and each of you has the opportunity to shape that connection with creativity and responsibility.

This edition highlights the growth of global technology firms and explores trending technologies that are shaping the modern digital landscape. It also reflects on the rise of startups, including inspiring journeys from emerging Indian startups, showing how new ideas can transform industries and create meaningful impact.

The insights shared by our alumni provide valuable perspectives for students who aspire to build successful careers in technology. Alongside this, the ideas and projects presented by our students demonstrate initiative, innovation, and the confidence to think beyond boundaries.

The edition reflects our commitment to developing professionals who are technically strong, ethically grounded, and ready to solve real-world challenges. Our students are not merely following change; they are contributing to it.

I sincerely thank the editorial team and every contributor for their hard work and commitment in bringing this vision to life.

योग: कर्मसु कौशलम् - Excellence in action is true yoga.  
Let us continue to learn, build, and lead with purpose.



**Dr. Vibha D. Patel**  
(HOD, IT)

# Index

Growth of Global Tech Firms

Perplexity

01

02

Trending Technologies:

Cursor | Google Antigravity

The Rise of Startups

Swish | Sarvam AI | Lucidity | Convin | Neysa

03

04

Insights from Alumni

By Dhruv Shah & Dharmit Shah

Ideas in Action

SkillScript | CampusConnect | ExamEcho

05

06

Aptitude & Puzzle  
Question

# Perplexity

While most "AI tools" are simply API wrappers, Perplexity is a massive engineering feat of real-time data orchestration. It doesn't just "know" things; it builds a fresh knowledge graph for every query you send. If you want to understand how it actually works, you have to look past the chat box and into the three-layered engine driving the platform.



Img source - USAII

## 1. The Real-Time Crawler: Exabyte-Scale Indexing

Traditional LLMs have a "knowledge cutoff" ; they are frozen in time. Perplexity bypasses this by running an AI-first search index. Imagine a hyper-efficient, constantly updated library that can scan new books the moment they're published. This is their advanced web crawler at work.

**The Scope:** Their crawler tracks over 200 billion unique URLs, far beyond what traditional search engines index for real-time relevance.

**The Hardware:** The indexing fleet runs on tens of thousands of CPUs and hundreds of terabytes of RAM, continuously processing new information.

**The Logic:** Instead of simple keyword matching, Perplexity uses ML-driven prioritization to decide which parts of the internet to "keep hot" (cached and ready) based on authority and topic relevance. This ensures that the most credible and up-to-date sources are always at hand.

## 2. The Retrieval-Augmented Generation (RAG) Pipeline

This is where the magic happens. When you type a query, the system doesn't just ask an LLM to guess. It follows a multi-stage Retrieval and Ranking process to find and synthesize information from its vast index.

**Hybrid Retrieval:** It simultaneously performs Lexical search (matching exact words and phrases) and Semantic search (understanding the underlying meaning and intent of your query via vector embeddings). This dual approach ensures both precision and relevance.

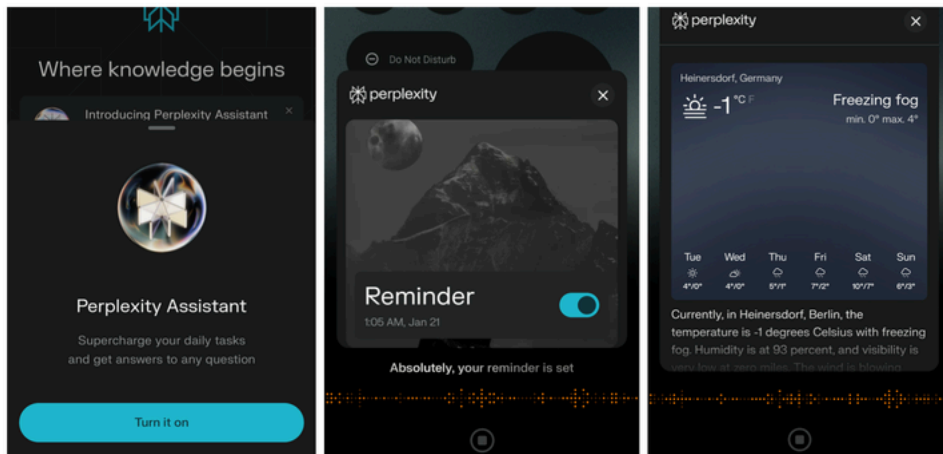
**The Reranker:** It pulls a massive set of "candidate" links that might be relevant, then uses a sophisticated Cross-Encoder model to rank them. This step is crucial for filtering out noise and ensuring that only the most high-fidelity, authoritative sources make it to the final stage.



Img source - Perplexity

**Prompt Stuffing:** The system then "stuffs" the top-ranked, verified search results into the LLM's context window (up to 128k–200k tokens or more). This process effectively "grounds" the LLM, forcing it to generate an answer based only on those verified facts, significantly reducing the risk of hallucinations.

Feature	Technical Specification
<b>Core Architecture</b>	<b>RAG (Retrieval-Augmented Generation)</b>
<b>Search Index</b>	200B+ proprietary indexed URLs (Real-time)
<b>Primary Models</b>	Sonar (Llama 3.1 based), GPT-4o, Claude 3.5, Gemini Pro
<b>Special Features</b>	Pro Search (300+ sources/query), Pages (AI-authored reports)
<b>Latency/Speed</b>	~2-3 seconds (Quick), 15-20 seconds (Deep Research)
<b>Accuracy</b>	92-97% factual grounding (via verifiable citations)
<b>Market Stats</b>	45M+ Monthly Active Users; #2 market is India
<b>Pricing</b>	Free (Basic)   \$20/mo (Pro)   \$40/mo (Enterprise)



Img source - Testing Catalog

### 3. The Inference Stack: Serving 20+ Models at Scale

Serving millions of queries monthly with real-time performance requires an elite, highly optimized infrastructure. Perplexity doesn't just use one model; they intelligently route your query to the best tool for the

job. **The Brain:** They utilize a "classifier model" (a smaller, faster LLM) to instantly detect your intent. If you're asking for code, it might route you to a model optimized for programming tasks like Claude 3.5 Sonnet; for a quick fact, it could use their highly efficient in-house Sonar model (based on Llama 3.1 or later). **GPU Orchestration:** Their backend leverages a massive fleet of NVIDIA H100 Tensor Core GPUs, managed by sophisticated Kubernetes clusters for dynamic scaling and resource allocation. **Optimization:** They employ NVIDIA Triton Inference Server and TensorRT-LLM to parallelize the "prefill" (processing your input query) and "decode" (generating the output answer) phases. This advanced optimization is precisely why you see the answer stream almost instantly, rather than waiting for long processing times.

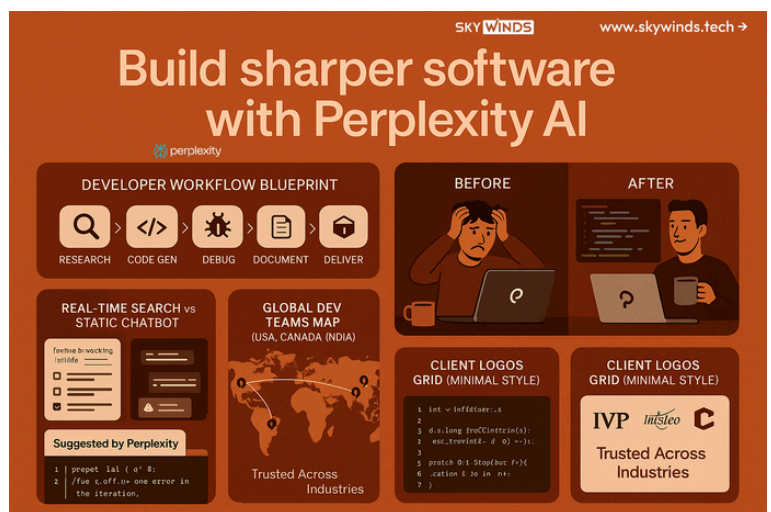
Component	Technology/Approach Used
Cloud Infrastructure	AWS (SageMaker HyperPod, EC2, various managed services)
Inference Server	NVIDIA Triton Inference Server, TensorRT-LLM
Base LLM Models	Llama 3.1, GPT-4o, Claude 3.5 Sonnet, Gemini Pro (and their proprietary Sonar models)
Search Logic	Retrieval-Augmented Generation (RAG) with advanced reranking
Backend Framework	Highly distributed systems using Python, Ray, and custom low-latency services
Data Indexing	Proprietary AI-first web crawler with ML-driven prioritization

For IT students, Perplexity is a masterclass in System Design. It proves that the future of AI isn't just about having the biggest model, it's about how efficiently you can connect that model to the real world. They solved the "Hallucination Problem" not through better math, but through better Information Retrieval.

### Perplexity AI: The Research Architect

The "Academic Focus" Pivot: Don't just search the web. Use the Focus button and select Academic. This forces Perplexity to pull from peer-reviewed journals (arXiv, PubMed, IEEE) rather than SEO-optimized blogs. It's a cheat code for your final year projects.

The "Collections" Instructions: Create a Collection for a specific project (e.g., "Deep Learning Research"). In the Collection settings, add Custom Instructions like: "Always output code in Python 3.12" or "Always prioritize research published after 2024." Every thread in that collection will now automatically follow those rules.



Img source - Skywinds Solutions

File Interrogation: Drag a 50-page PDF research paper into the search bar. Ask: "What are the three biggest limitations in this author's methodology?" Perplexity will find the "fine print" in seconds.

### Resources & Further Reading

- Perplexity Engineering Blog: [High-Performance GPU Memory Transfer](#)
- NVIDIA Case Study: [Scaling LLM Inference to 400M+ Queries](#)
- Vespa.ai: [How Perplexity built an AI Search Engine](#)

# THE AI-FIRST IDE: Cursor & The "Shadow" Logic of Autocomplete

If you think Cursor is just VS Code with a ChatGPT plugin, you're missing the most interesting engineering happening in the IDE space today. Cursor's magic doesn't come from the LLM; it comes from the Local Context Engine.

## 1. The Merkle Tree Indexing

When you open a folder in Cursor, it doesn't just read your files it indexes them into a Merkle Tree. This allows the IDE to track changes at the byte level.

## 2. Speculative Decoding: The Speed Secret

Ever notice how Cursor's "Tab" suggestions feel faster than your brain? That's Speculative Decoding.

The Logic: Cursor runs a small, 7B-parameter "draft" model locally on your machine (or a fast edge server) to guess the next few lines.

The Validation: A larger "frontier" model (like Claude 3.5 or GPT-4o) then verifies those guesses in the background. If the small model was right, the text appears instantly. If it was wrong, it corrects it before you even see the flicker.

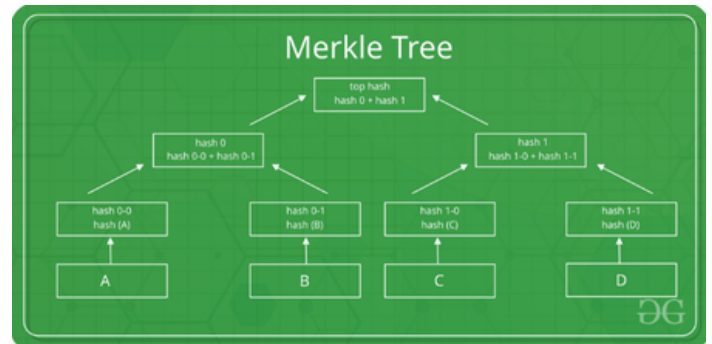


Image Source: <https://www.geeksforgeeks.org>

The Tech: It uses Tree-sitter to parse your code into Abstract Syntax Trees (ASTs). The Result: Instead of sending your whole file to an AI (which is slow and expensive), Cursor sends only the relevant "branches" of your code logic.

## 3. Cursor: The Speed-Coding Master

The @ Menu (Context Injection): The most powerful key in Cursor is @.

@Files: Give the AI the exact file you're working on.

@Web: Let the AI search the latest documentation (e.g., the 2026 Next.js updates) while it writes your code.

@Codebase: Ask questions about your entire project structure, not just the open tab.

CMD + K (The Inline Surgeon): Highlight a messy function and hit CMD + K. Type: "Refactor this for O(n) complexity and add JSDoc comments." It rewrites the code right in front of you.

The .cursorrules File: This is a secret file you can add to your root directory. Write your project's styling rules here (e.g., "Use Tailwind utility classes only"). Cursor will now stop suggesting old-school CSS and stick to your project's vibe.

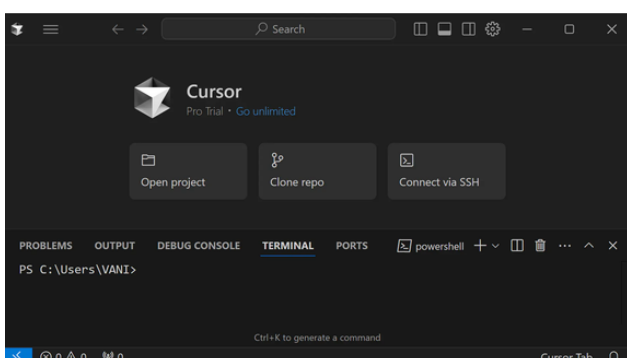


Image Source : <https://www.testingdocs.com/cursor-ai-ide-features/>

# FROM EASTER EGG TO AGENT: The Evolution of Google Antigravity

In the early days of the web, Google Anti-Gravity was a legendary Chrome Experiment by Ricardo Cabello. It used a Box2D physics engine to let users "tumble" the Google homepage. But in 2026, Google has reclaimed the name for something far more powerful: The Antigravity AI IDE.

## The Tech Behind the "Chaos"

The original experiment was a masterclass in DOM Manipulation. Each HTML element (the logo, search bar, buttons) was mapped to a rigid physics body.

Library: It primarily used Matter.js or Box2D to calculate collisions and gravity in the browser's main thread.

Modern Rebirth: Today, Google has used the "Antigravity" branding for its Agentic IDE (powered by Gemini 3).

## The "Agent-First" Stack

The new Antigravity IDE isn't just an editor; it's a Mission Control for AI agents.

Browser Actuation: Unlike other IDEs, Antigravity agents can open a headless Chrome instance, navigate to your localhost, and "see" if the UI they just coded actually works.

Manager View: It uses a "parallel orchestration" layer where you can dispatch five different agents to solve five different bugs across your repo simultaneously.

## Comparison: The Tech Stack Matrix

Feature	Cursor (The Challenger)	Google Antigravity (The Titan)
Core Model	Model-Agnostic (Claude/GPT/Llama)	Gemini 3 / Gemini 3.1 Pro
Indexing	Local Vector Embeddings (RAG)	Google Knowledge Graph + Live Web
Unique Tech	Speculative Decoding (High Speed)	Browser Actuation & Multi-Agent View
Base Platform	VS Code Fork	VS Code Fork (Heavily modified)

## Google Antigravity: The Agent Pilot

Plan Mode vs. Fast Mode: For big architectural changes, always use Plan Mode. The AI will generate a "flight plan" first. Review it, delete the steps you don't like, and then hit "Execute."

Browser-in-the-Loop Debugging: If your UI looks weird, tell the Antigravity agent: "Open the built-in browser, go to localhost:3000, and tell me why the header is overlapping." The agent will "see" the DOM and fix the CSS automatically.

Artifact Comments: Antigravity generates "Artifacts" (like task lists or code diffs). You can comment on these exactly like a Google Doc. If the AI's plan is 80% right, just highlight the 20% that's wrong and add a comment: "Actually, use PostgreSQL for this instead of MongoDB."

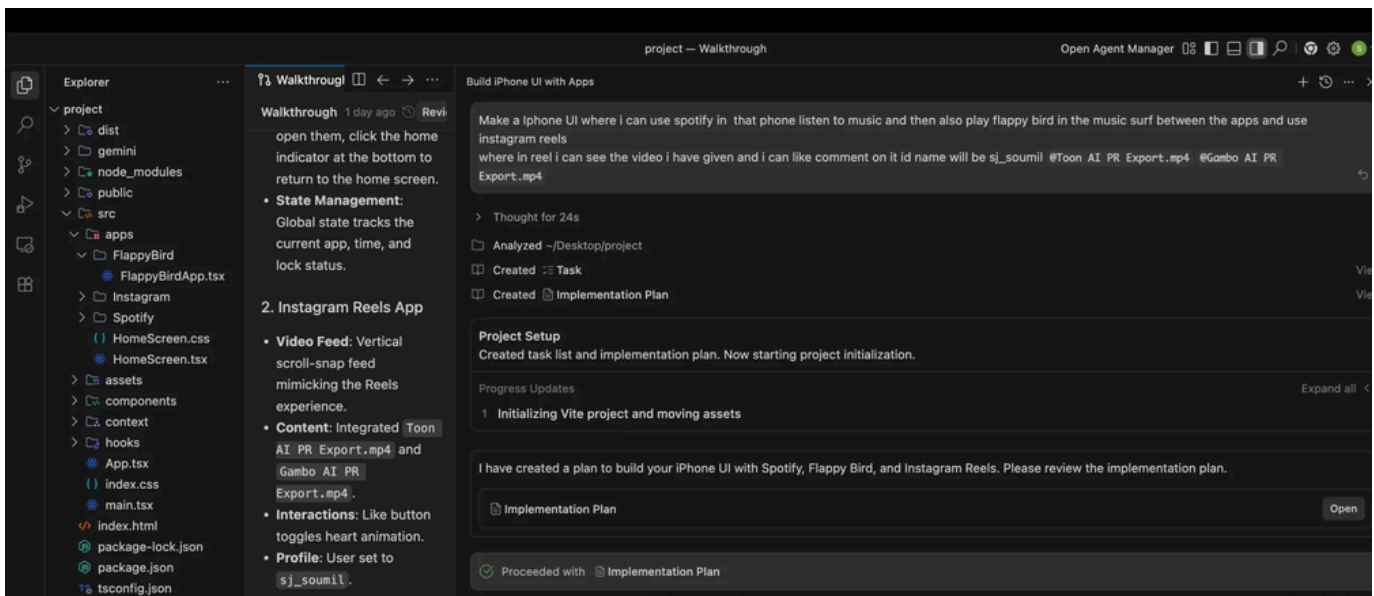


Image Source : <https://www.testingdocs.com/cursor-ai-ide-features/>

# Cursor Vs Antigravity

Feature	Cursor IDE (The Precise Editor)	Google Antigravity (The Agent Pilot)
<b>Mental Model</b>	Assist-First: Enhances manual coding	Agent-First: Automates entire tasks
<b>Key Philosophy</b>	"Vibe Coding" with tight human control	"Mission Control" delegation
<b>Parallelism</b>	Up to 8 parallel agents in git worktrees	Massive multi-agent orchestration grid
<b>Browser Integration</b>	Embedded Chromium for DOM inspection	Full Browser Actuation (Auto-testing)
<b>Unique Tech</b>	Merkle Tree indexing & Speculative Decoding	Artifacts System (Logs, Recordings, Plans)
<b>Security</b>	Sandboxed terminal & Privacy Mode	Terminal Sandboxing & Google-level SSO
<b>Best For...</b>	Production apps & established codebases	Rapid prototyping & "Greenfield" projects
<b>Current Status</b>	Market Leader (Battle-tested)	Public Preview (Experimental/Cutting-edge)

## Resources & Courtesy

Cursor Engineering: [How we built the fastest autocomplete](#)

Google Research: [Agent-First Development with Gemini 3](#)

Original Anti-Gravity Demo: [Mr. Doob's Google Gravity](#)

# The Rise of Start-ups

## Swish: Engineering Speed in Bangalore's Food-Tech Race

In Bengaluru, a city defined by startups, software, and long workdays, convenience is key. Swish, founded in 2024 by Aniket Shah, Ujjwal Sukheja, and Saran S, aims to be a technology-driven food delivery platform for urban consumers who value speed over discounts. Instead of simply competing on price with major players like Zomato and Swiggy, Swish emphasizes operational accuracy. The goal is to provide a quicker and more reliable delivery experience in densely populated areas.

Swish operates as a real-time logistics engine. The platform uses live tracking, smart order distribution, and route planning to shorten delivery times. Rather than assigning riders at random, the system considers factors like traffic, kitchen prep times, and local demand before sending out an order. This approach lowers idle time and boosts reliability, which is important for frequent urban ordering.

In addition to logistics, Swish uses demand data to improve backend efficiency. By looking at peak-hour trends and customer habits, it can predict demand in various neighborhoods. This helps with smarter staffing and inventory choices for partner kitchens. In a business with slim margins, this data-driven approach cuts down on waste and improves overall cost effectiveness. Swish represents a larger trend in food technology: delivery is evolving from just transporting meals to using data to boost speed and scale.

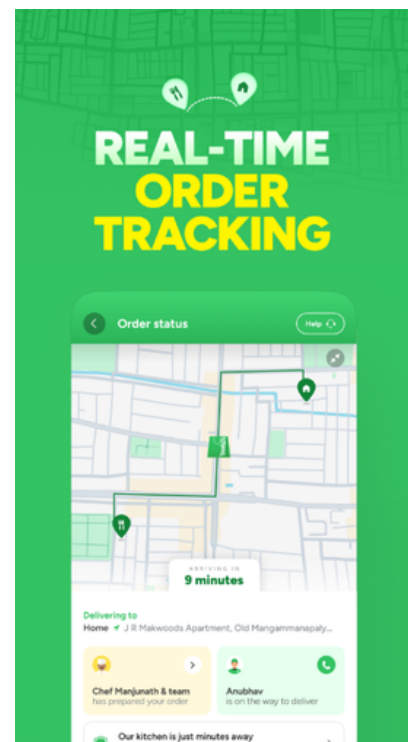


Image Source : Swiss-PlayStore



Image Source : <https://economictimes.indiatimes.com/>

Founder:

Aniket Shah, Ujjwal Sukheja and Saran S.

## Sarvam AI: Building India's Sovereign AI Stack

As artificial intelligence becomes key to economic growth, the focus is shifting from simply using AI to owning the underlying technology. Sarvam AI, founded in July 2023 by Vivek Raghavan and Pratyush Kumar, is based in Bengaluru. The company aims to create a complete AI ecosystem tailored to India's languages and regulations. Instead of relying solely on global, one-size-fits-all models, it is developing generative AI systems specifically designed for Indian users and institutions.

Sarvam AI's approach centers on multilingual large language models optimized for Indian languages. With hundreds of languages and dialects spoken across the country, many global models struggle with context and regional diversity. Sarvam tackles this by training its models on local datasets, which leads to more accurate text generation, speech-to-text capabilities, and conversational AI across various Indian languages. This localized training makes AI more accessible beyond English-speaking urban users and reaches broader demographic groups.

Technologically, Sarvam AI operates across the entire stack, from developing foundational models to building deployment infrastructure and enterprise-ready APIs. Its cloud-native platforms focus on secure and compliant deployment, allowing businesses and public institutions to integrate AI into their workflows while following data governance standards within India. In this way, Sarvam AI is part of a larger shift in India's tech evolution, moving from being mainly an AI consumer to becoming a builder of essential AI infrastructure.

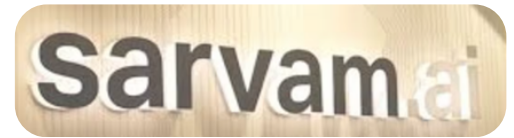
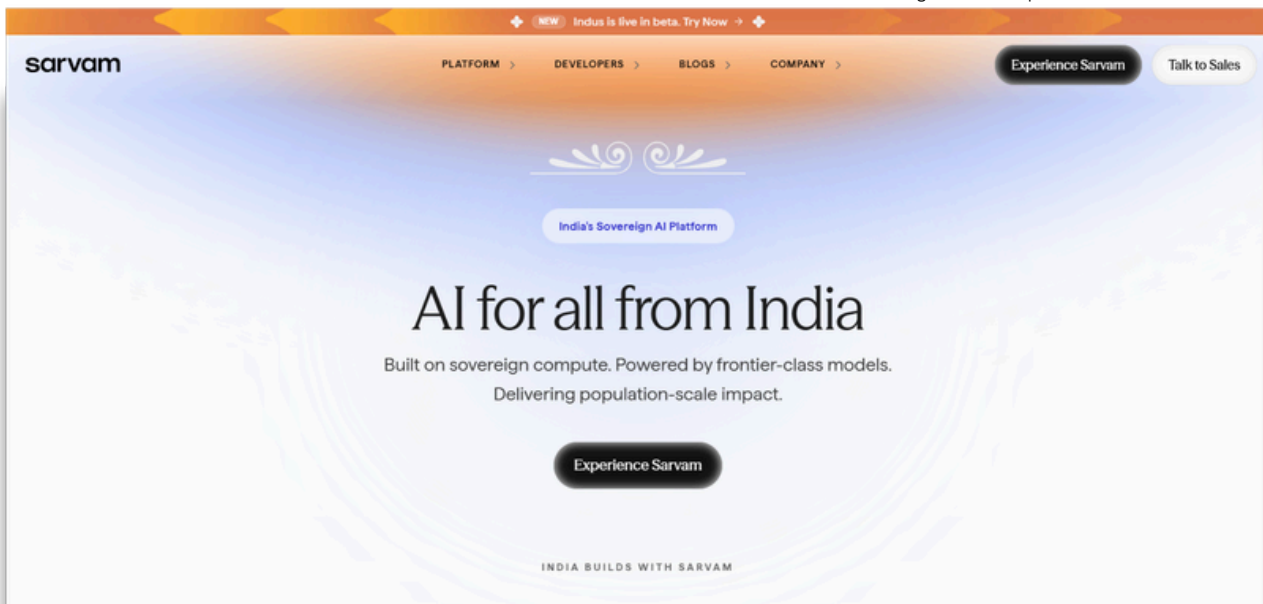


Image Source : <https://economictimes.indiatimes.com/>

Founder:  
Dr. Vivek Raghavan &  
Dr. Pratyush Kumar

Image Source : <https://www.sarvam.ai>





## Lucidity: Engineering Efficiency in the Cloud Era

As businesses speed up their use of the cloud, infrastructure costs, especially for block storage, have quietly drained IT budgets. Founded in 2021 by Nitin Singh Bhadauria and Vatsal Rastogi, and based in Bengaluru, Lucidity aims to tackle this issue. The startup creates a smart storage management layer that helps organizations cut waste and manage cloud spending without changing their existing applications or workflows.



Image Source : <https://economictimes.indiatimes.com/>

Founder:  
Vatsal Rastogi &  
Nitin Singh Bhadauria

The main feature of Lucidity's platform is the automated, real-time scaling of cloud block storage. In traditional setups, teams often overprovision resources to avoid outages. This leads to unused capacity and higher bills. Lucidity shifts this reactive approach to one that uses rules to automatically adjust storage volumes based on current demand. By tracking usage patterns and reallocating capacity in real time, the system makes sure storage increases or decreases exactly when needed, improving both cost efficiency and performance predictability.



Image Source : <https://www.lucidity.cloud/>

Working at the intersection of cloud-native architecture, analytics, and automation, Lucidity introduces a “NoOps” method to storage management. For DevOps and FinOps teams managing complex, multi-cloud systems, this changes infrastructure from a fixed cost to an optimized, self-regulating system. In the wider SaaS market, Lucidity highlights an important fact of the cloud era: scaling is easy; the real innovation is in smart scaling.

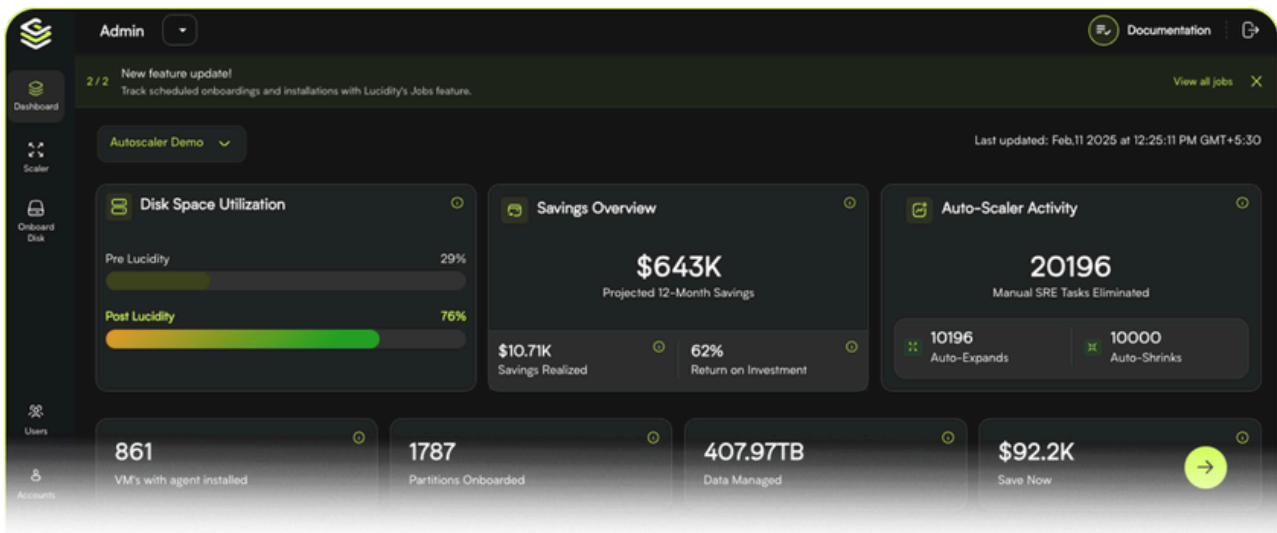


Image Source : <https://www.lucidity.cloud/>



## Convin: Turning Conversations into Competitive Advantage

In an experience-driven economy, customer conversations are more than just routine interactions. They are structured data waiting to be decoded. Convin is an AI-powered conversational intelligence platform that helps sales, support, and collections teams turn everyday calls into measurable performance insights. Instead of relying on manual call sampling or random quality checks, the platform automates large-scale analysis. This brings consistency and depth to monitoring conversations.



Image Source : <https://yourstory.com>

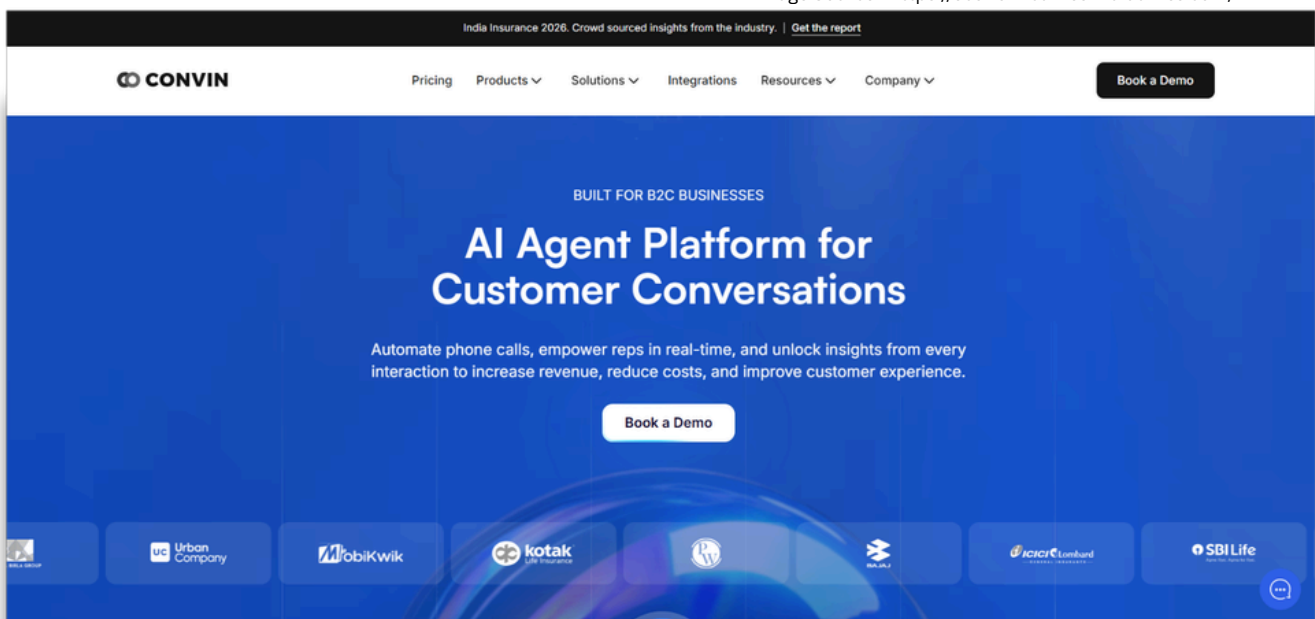
Founder:

Ashish Santhalia, Atul Shree  
Bharat Patidar, Durgesh Kumar

Convin uses speech recognition, Natural Language Processing (NLP), and machine learning to transcribe and evaluate interactions in real time. It tracks changes in sentiment, compliance with rules, patterns in handling objections, and sales cues. This transforms unstructured voice data into useful metrics. Managers can view team-wide performance dashboards, while agents get targeted coaching based on data. This turns improvement into an ongoing feedback loop instead of a periodic review.

What sets Convin apart is its ability to connect conversational behavior with business results. By identifying patterns from top agents, the system highlights techniques that directly impact conversions and customer satisfaction. In busy settings like contact centers, this shifts AI from simply monitoring to actively improving performance. It gives organizations a structured way to turn conversations into a lasting competitive advantage.

Image Source : <https://economictimes.indiatimes.com/>



## Neysa: Accelerating the Infrastructure Behind Generative AI



As generative AI models become larger and more demanding in terms of computing power, infrastructure has become a key factor for success. Neysa positions itself as a cloud and acceleration platform focused on AI, specifically for training and deploying GenAI systems. The company does not provide generic cloud services. Instead, it focuses on high-performance environments that are optimized for large language models and multimodal workloads.

At its core, Neysa offers GPU-powered compute clusters, high-speed networking, and scalable storage designed for parallel processing and fast data movement. Training advanced AI models requires massive computing power, which traditional cloud setups often struggle to provide. By combining hardware acceleration with optimized orchestration and workload scheduling, Neysa allows for quicker training cycles and smoother deployment of AI applications.

Neysa operates at the intersection of AI computing and cloud architecture. It simplifies backend complexity through containerized environments and automated scaling. This enables startups and businesses to focus on model development instead of managing infrastructure. In the changing AI landscape, Neysa represents an important shift. Breakthrough AI is no longer just about algorithms; it also depends on access to powerful, scalable, and well-designed computing resources.



Image Source : <https://www.linkedin.com/>

Founder:  
Sharad Sanghi

Image Source : <https://neysa.ai/>

A screenshot of the Neysa website's landing page. At the top, there is a purple navigation bar with the Neysa logo on the left and two buttons, "Contact Sales" and "Request a Demo", on the right. Below the navigation bar, the main heading reads "Neysa Velocis: Your AI Acceleration Cloud System." followed by the sub-headline "Build, Deploy, and Scale AI - Simplified." A short paragraph describes the service: "Go from concept to production faster with our integrated AI Acceleration Cloud System. Train, test, deploy, and monitor your AI models, all from a single dashboard, without the complexity of managing multiple tools." A prominent purple button labeled "Discover Neysa Velocis" is positioned below the text. On the right side of the page, there is a 3D graphic of colorful, translucent cubes in shades of blue, purple, and red, arranged in a cluster with some floating around it.

# Insights From Alumni

The software industry is evolving at an extraordinary pace. Modern applications are no longer built merely to function; they are expected to be fast, scalable, secure, search-engine friendly, and increasingly intelligent. Users today demand seamless digital experiences, businesses demand performance and scalability, and developers must deliver both. To meet these growing expectations, modern frameworks such as Next.js and NestJS have emerged as powerful solutions for building production-ready full-stack applications. At the same time, Artificial Intelligence (AI) is transforming how applications are designed and how users interact with them.

Together, these technologies represent a significant shift in how modern software systems are architected and deployed.

## **Next.js: Redefining Frontend Performance and User Experience**

Frontend development has evolved significantly over the last decade. While React revolutionized component-based UI development, large-scale applications began facing challenges related to performance and search engine optimization (SEO). Traditional client-side rendering sometimes resulted in slower initial load times and reduced visibility on search engines.

Next.js addresses these challenges by enhancing React with powerful features such as Server-Side Rendering (SSR), Static Site Generation (SSG), and Incremental Static Regeneration (ISR). These rendering strategies allow pages to be pre-rendered before reaching the user, significantly improving performance and SEO rankings.

In practical terms, this means websites load faster, provide smoother user experiences, and rank better on search engines. In today's competitive digital landscape, even a small delay in loading time can impact user retention and business growth. Next.js helps mitigate these issues through automatic code splitting, optimized image handling, and built-in performance enhancements.

As businesses increasingly prioritize speed and performance, Next.js has become a preferred choice for startups, enterprises, and large-scale platforms seeking to deliver optimized digital experiences.

## **NestJS: Structured Backend for Scalable Systems**

On the backend side, scalability and maintainability are critical factors in modern application development. While Node.js and Express provide flexibility and simplicity, larger projects often struggle with maintaining clean architecture as they grow.

NestJS was introduced to solve this problem by bringing structure and enterprise-grade architecture to Node.js development. Built with TypeScript at its core, NestJS promotes modular design, dependency injection, and separation of concerns. These architectural principles ensure that applications remain organized, testable, and scalable over time.

In large-scale systems where multiple developers collaborate, having a well-defined project structure is essential. NestJS encourages best practices by dividing applications into modules, controllers, and services, making codebases easier to manage and extend.

As organizations increasingly adopt microservices and distributed architectures, frameworks like NestJS provide the foundation for building reliable backend systems capable of handling high traffic and complex business logic.

## **Artificial Intelligence: The Intelligent Layer of Modern Applications**

Beyond frontend and backend frameworks, Artificial Intelligence is rapidly becoming an integral part of modern software systems. AI is no longer confined to research environments; it is now embedded directly into web and mobile applications.

Modern applications frequently incorporate AI-powered features such as chatbots, recommendation engines, predictive analytics, intelligent search systems, and automated workflows. These features enhance user engagement by delivering personalized experiences and data-driven insights.

Generative AI, in particular, has introduced new possibilities. Applications can now process natural language, generate content, summarize information, and automate repetitive tasks. Businesses are leveraging AI to improve customer service, optimize operations, and enhance decision-making processes.

AI is also transforming the development workflow itself. AI-powered coding assistants help developers write code faster, detect errors efficiently, and explore optimized solutions. This shift increases productivity and allows engineers to focus more on system design and problem-solving rather than repetitive coding tasks.

## **The Convergence of Performance, Structure, and Intelligence**

The true transformation in modern software development lies in the convergence of performance-focused frontend frameworks, structured backend architectures, and AI-driven capabilities. Applications today must handle growing user bases, manage large volumes of data, and provide real-time responses all while maintaining reliability and security.

Cloud computing, DevOps practices, containerization, and serverless technologies further enhance this ecosystem by enabling continuous deployment and horizontal scalability. In such environments, structured frameworks like NestJS and optimized frontend solutions like Next.js become essential tools.

One reality of the IT field is that change is constant. New frameworks, new tools, new AI capabilities, and new development practices emerge every day. What is trending today may evolve tomorrow. The demand in the IT industry continues to evolve, and as professionals, we must match our pace with innovation. Continuous learning is no longer optional; it is necessary.

Wishing you all the very best for your future. Stay curious, stay consistent, and keep building.

**~ Dhruv Shah (2021-2025 batch)**

## **Beyond the Lone Model: Welcome to the LLM Council**

For a long time, our interaction with artificial intelligence followed a simple, linear path. You had a question, you picked your preferred Large Language Model (LLM) be it GPT-4, Claude, or Gemini and you accepted its answer as the definitive word. This "lone architect" model was powerful, but flawed. It was susceptible to hallucinations, subtle biases embedded in a single dataset, and the limitations of a solitary reasoning path.

We are now entering a new era. The industry is shifting from trusting a single "black box" to embracing a multi agent orchestration framework.

It's time to meet the LLM Council.

## What is the LLM Council? A Shift in Architecture

Populated by leading AI researchers, and recently highlighted by figures like Andrej Karpathy, the LLM Council is a decentralized deliberation framework. It replaces the single model system with a collaborative committee. Instead of relying on one super intelligence, we query multiple frontier models simultaneously, allowing their collective intelligence to filter out individual errors and biases. The goal isn't just to generate an answer; the goal is to derive the most robust, cross-verified consensus answer possible. It is a system built on democratic principles, peer review, and adversarial testing.

### The Architecture: Triangulating the Truth in Three Stages

How does this actually work? The LLM Council operates through a disciplined, multi-stage pipeline designed to minimize error.

#### Stage 1: Independent Deliberation (Preventing Groupthink)

The moment a user submits a complex prompt, it is dispatched in parallel to a panel of distinct LLMs for example, GPT-4, Claude 3.5, and Gemini 1.5. Crucially, these models do not communicate. Each model develops its own comprehensive response and reasoning trace based solely on its own pre-training data. This foundational step ensures a maximum diversity of thought and interpretation, essential for finding the truth.

#### Stage 2: Anonymous Peer Review (Critique and Ranking)

Once the independent drafts are complete, the system enters the critical peer review phase. The original prompt and all anonymized candidate responses are sent back to the same set of councilors. Each model must now critique, compare, and rank the anonymous responses, including its own, which it does not recognize. Models are exceptionally strong critics; they can often spot logical inconsistencies or factual errors in other models' text that they might miss in their own generation.

#### Stage 3: The Chairman's Synthesis (Final Assembly)

The final step involves the "Chairman" often a different model specialized in synthesis or summarizing complex debates. The Chairman receives the original prompt, the full set of anonymized candidate answers, and the detailed peer critiques and rankings. Its job is to generate a single, consolidated, high-quality final answer that incorporates the best insights while explicitly discarding any outliers, hallucinations, or contradictions identified during the deliberation process.

## Why You Need an LLM Council: High Governance AI

Implementing an LLM Council offers several critical benefits for applications that demand high reliability:

- 1. Hallucination Defense:** If one model hallucinates, the other councilors will almost certainly flag the error during peer review.
- 2. Mitigated Bias:** The consensus process filters out unique training data biases. What remains is universally verified information.
- 3. Future-Proofing:** Your AI stack is modular. You can easily swap new models (e.g., Llama 3) into a "seat" on the council without rewriting your application logic.

While running multiple calls increases per query costs and latency, the investment buys invaluable confidence. We are moving from AI as a fancy chatbot to AI as a reliable governance tool.

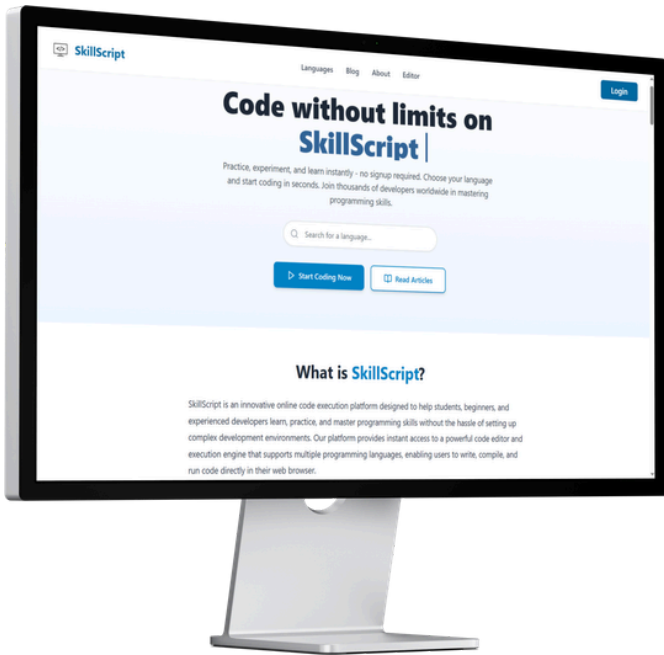
# Project Synopsis



*A structured approach to understanding the problem, designing an effective solution, and implementing it using appropriate technologies.*

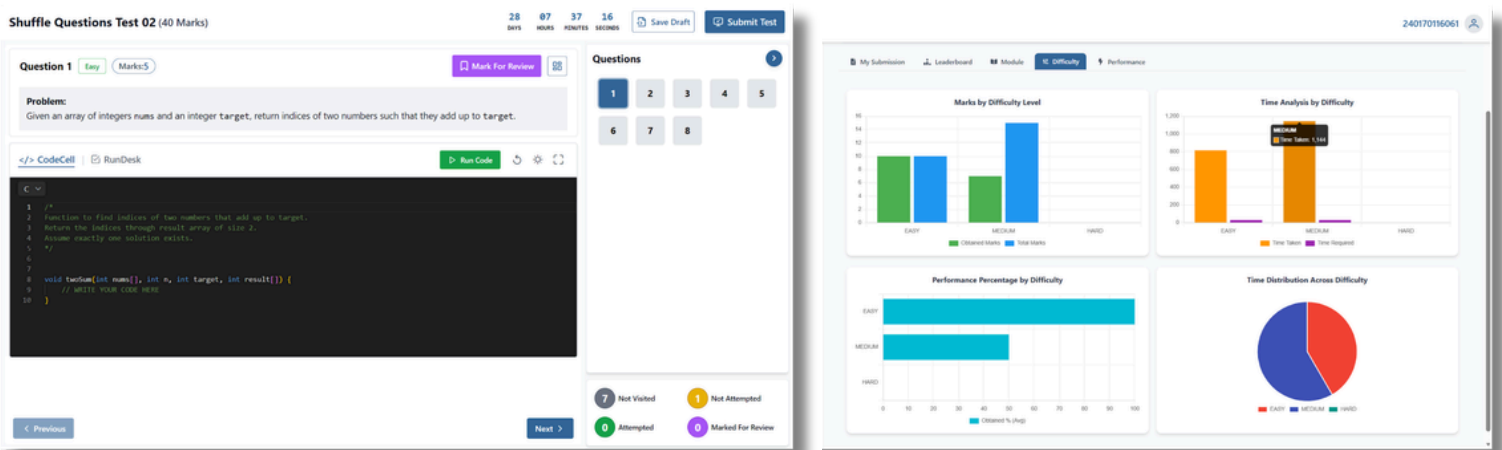
# 1. SkillScript

~MERN Stack, Docker, SEB, Gemini API



SkillScript is a comprehensive academic programming practice and assessment platform designed to enhance practical learning and streamline evaluation processes in educational institutions. It serves as a centralized system where students can practice programming tasks efficiently and faculty members can manage academic assessments digitally. The platform enables faculty to create and maintain a structured Question Bank, design tasks such as tests and lab practicals, and assign them to students. The platform supports multiple programming languages - C, C++, Java, Python, HTML/CSS/JavaScript

Each task can be aligned with specific subject modules and evaluation rubrics. Faculty members have the flexibility to create customized rubrics and associate them with tasks for structured and outcome-based evaluation. Additionally, subjects and their respective modules are dynamic, allowing institutions to define, add, or modify them according to curriculum needs and university regulations. The organization of subjects along with their respective modules supports and promotes outcome-based learning by aligning assessments with defined learning objectives.



SkillScript uses Permission-Based Access Control (PBAC) to manage system operations through granular permissions. Users are assigned specific rights such as creating assessments, publishing results, configuring rubrics, submitting code, reviewing submissions, viewing analytics, and generating reports. Access to these actions is securely validated using JWT-based authentication.

A distributed client–server architecture has been designed and implemented using the MERN stack to separate concerns across different layers of the system:

- Frontend Layer: Provides an interactive browser-based coding environment using ReactJS
- Backend Layer: Node.js and Express.js manage APIs, authentication, business logic, and evaluation workflows
- Execution Engine: Docker-based sandboxed containers are used to compile and run user code securely
- Database Layer: MongoDB is used for storing user data, submissions, rubrics, analytics, and assessment reports

The execution engine has been developed using a containerized microservice approach where each code submission is executed within an isolated runtime container to prevent interference with the host system.

Rank	Name	Enrollment	Marks	Status	Time Taken	Submitted At
1	DHRUMIBEN SACHINBHAI TAILOR	240170116070	25/25	Present	17:43	20 Aug 2025, 15:45:43
2	DARSHIL PARESHBHAI KAKADIYA	240170116033	25/25	Present	18:52	20 Aug 2025, 15:44:04
3	HET SHAILESHBHAI KAPADIYA	240170116027	25/25	Present	42:22	20 Aug 2025, 16:25:25
4	MANTHAN SANJAYKUMAR DOBARIYA	240170116015	23/25	Present	58:45	20 Aug 2025, 15:43:08
5	ANAND MAHENDRASHEEL UPASAK	240170116074	23/25	Present	54:40	20 Aug 2025, 15:20:01
6	KRISHKUMAR PRAVINBHAI RAMANI	240170116058	20/25	Present	30:22	20 Aug 2025, 15:23:22
7	PARAM VIJAYBHAI RADADIYA	240170116057	20/25	Present	41:01	20 Aug 2025, 15:23:28
8	VIDHI VISHNUKUMAR SINDHI	240170116066	20/25	Present	1:00:02	20 Aug 2025, 15:22:33
9	RUSHI JIGNASUBHAI VYAS	240170116078	19/25	Present	59:37	20 Aug 2025, 15:25:28
10	JENI PANKAJKUMAR THESEYA	240170116029	19/25	Present	1:00:02	20 Aug 2025, 15:28:56
11	FIRTH BHAVINKUMAR SHAH	240170116063	18/25	Present	58:38	20 Aug 2025, 15:34:30
12	DHRUVY SAKSHIKUMAR PATIL	240170116047	18/25	Present	1:00:02	20 Aug 2025, 15:26:19
13	SALUNI KISHORIDAS NAINJLE	240170116041	18/25	Present	1:00:02	20 Aug 2025, 15:34:22
14	KHUSHBU ASHOKBHAI SAGATHIYA	240170116061	17/25	Present	36:54	20 Aug 2025, 16:07:39
15	MEET ASHISHBHAI ITALIYA	240170116028	17/25	Present	37:25	20 Aug 2025, 15:37:12
16	DEVALIYA JAYDIP BHAVESHBHAI	2025/T02D04	17/25	Present	58:34	20 Aug 2025, 15:28:39
17	ASHISH SINGH SLATHIA	240170116001	17/25	Present	58:51	20 Aug 2025, 15:38:30
18	DIMPAL BHAKHUBHAI ZORA	240170116030	17/25	Present	59:30	20 Aug 2025, 15:30:20
19	TANISHA AMITKUMAR JOSHI	240170116031	17/25	Present	1:00:02	20 Aug 2025, 15:30:32

**C Programming**

**Performance Summary**  
 Total Marks: 25 | Average Marks: 9.74 | Total: 88 students | Absent: 14 student(s) | Number Of Student(s) Obtained Full Marks: 3 student(s)

Student Name	Enrollment	Marks	Percentage
DHRUMIBEN SACHINBHAI TAILOR	240170116070	25/25	100.0%
DARSHIL PARESHBHAI KAKADIYA	240170116033	25/25	100.0%
HET SHAILESHBHAI KAPADIYA	240170116027	25/25	100.0%
MANTHAN SANJAYKUMAR DOBARIYA	240170116015	23/25	92.0%

SkillScript includes an auto-evaluation system that assesses student submissions based on predefined test cases. After task completion, both students and faculty can view results, marks, and performance analytics. It promotes paperless academic management by digitizing lab records and test evaluation workflows. The platform’s multi-level institutional architecture (Institute – Department – Faculty – Student) enables scalable deployment across multiple departments or institutions.

The platform integrates controlled coding environments, randomized question banks, and session monitoring to ensure integrity during lab exams making assessments more credible and transparent.

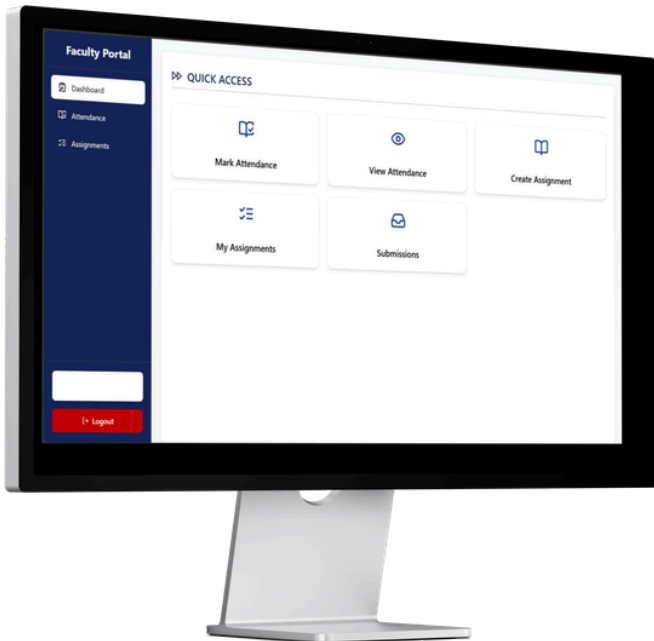
From conducting coding exams, storing student lab records securely, generating performance analytics, to automatically preparing academic reports—this system provides a complete digital ecosystem rather than a single-feature solution.



~ Dhruv Patel, Romil Patel, Jay Gohil,  
 Jaykishan Chandani, Khushi Patel,  
 Nazneen Patel, Nitigna Hihoriya, Neha Patel

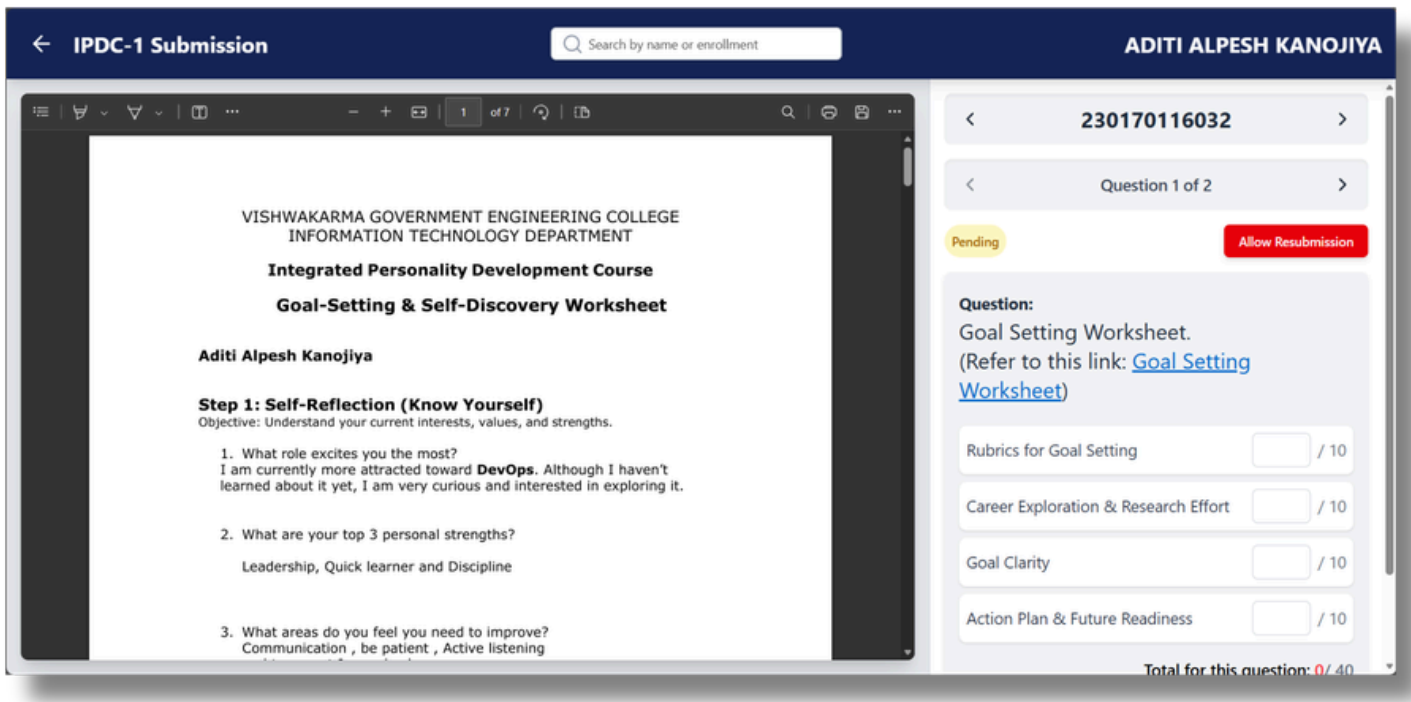
# 2. CampusConnect

~Node.js, MongoDB, JWT authentication, TanStack Query, and ShadCN UI



Campus Connect is redefining how educational institutions manage academics by bringing attendance tracking and assignment workflows into one powerful digital platform. Designed to replace error-prone, paper-based systems, Campus Connect offers a centralized solution where students, faculty, and administrators interact seamlessly. From real-time attendance visibility to structured assignment evaluations, the platform simplifies daily academic operations while promoting transparency and efficiency across campuses.

At the heart of Campus Connect lies its role-based architecture, carefully tailored for Students, Faculty, HODs, and Admins. Students can monitor their lecture and lab attendance, submit assignments online, and receive rubric-based feedback anytime, anywhere. Faculty members benefit from smart tools to record attendance, filter students by attendance percentage, design assignments with customizable rubrics, and evaluate submissions in a structured manner saving time while maintaining academic rigor.

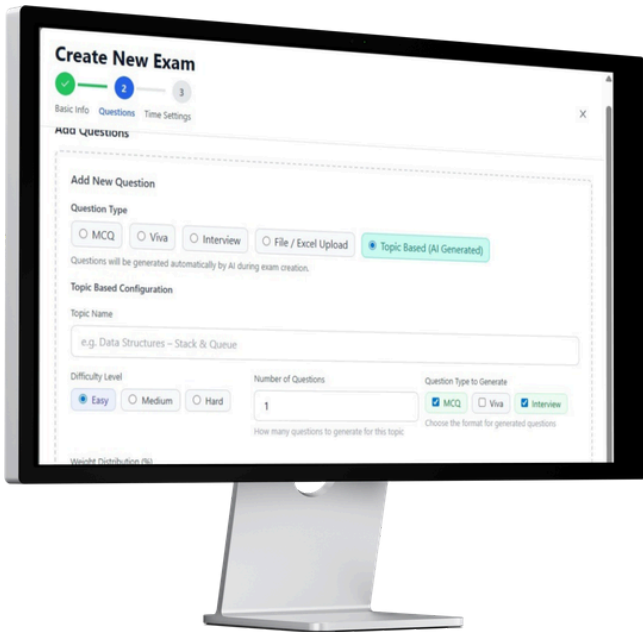


Faculty can access a student's assignment submission through an iFrame and perform the evaluation using defined rubrics.



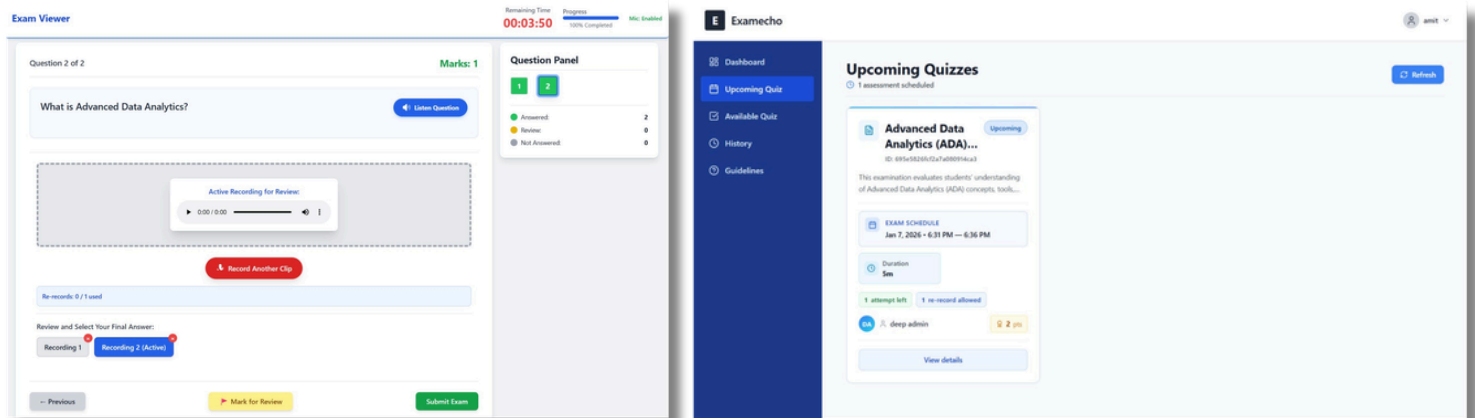
# 3. ExamEcho

~MERN Stack, FastAPI (Python), OpenAI & Gemini APIs, Redis, Clouinary



ExamEcho is an AI-powered examination portal developed to support teachers in designing, conducting, and evaluating viva voice, interviews, and theory-based examinations in a structured and efficient manner. The project was created to address challenges such as manual question preparation, inconsistent assessment standards, time-consuming evaluations, and subjective grading practices commonly faced by educators. The platform enables teachers to automatically generate syllabus-aligned, module-wise and topic-wise questions with adjustable difficulty levels.

This ensures uniform coverage of academic content, reduces repetitive effort in test creation, and helps maintain consistency across different batches, subjects, and evaluators. ExamEcho uses AI-driven evaluation to analyze student responses for correctness, conceptual understanding, clarity of explanation, and answer structure. Based on this analysis, the system provides objective scoring, detailed examiner-like feedback, and performance insights, allowing teachers to assess students fairly while significantly reducing manual workload.



In addition, the integration of Speech-to-Text and Text-to-Speech technologies allows realistic, interview-style examinations and smooth interaction during assessments. While the primary focus is on empowering educators with reliable and scalable evaluation tools, the platform also benefits students by offering transparent feedback and performance insights that support continuous improvement in understanding, communication, and confidence.

~ **Aryan Shah (Team Leader), Amit Ahir, Deep Prajapati, Harsh Shaileshkumar Patel, Murtaza Gangardiwala, Ghediya Purv Maheshbhai, Hingrajiya Vansh Kamleshbhai**



# Aptitude & Puzzle Questions



# Puzzle

1.

Three developers - Alice, Bob, and Carol - are working on different projects:

- Alice is not working on the mobile app
- The person working on the web app uses React
- Bob doesn't use React
- Carol is working on either the mobile app or the AI project Who is working on what?

2.

A system requires a 4-digit PIN.

Clues:

1. The sum of all digits is 20
2. No digit repeats
3. The first digit is twice the last digit
4. The second digit is greater than the third digit
5. The second digit is 5

What is the PIN?



3.

If A is the brother of B; B is the sister of C; and C is the father of D, how D is related to A?

- a) Brother
- b) Sister
- c) Nephew
- d) Cannot be determined

4.

What has six faces, but does not wear makeup, has twenty-one eyes, but cannot see? "What is it?"

5.

If in a certain code language:

- "CODE" is written as "DPEF"
- "TECH" is written as "UFDI" What will "DATA" be written as?

6.

A server can handle 100 requests per second. During peak hours, requests arrive at 120 per second. If this continues for 5 minutes, how many requests will be queued?

7.

A computer has 4 memory slots arranged in a row. Red, Blue, Green, and Yellow RAM sticks need to be inserted following these rules:

- Red must be in an end position
- Blue cannot be next to Green
- Yellow must be between two other colors

What is one valid arrangement?

# Aptitude



a)

```
#include <stdio.h>

int main() {
    int a = 6;
    int b = 0;

    while (a < 10) {
        a = a / 12 + 1;
        a += b;
    }

    printf("%d", a);
    return 0;
}
```

b)

```
#include <stdio.h>
void fun(int n) {
    if (n == 0)
        return;
    fun(n - 1);
    printf("%d ", n);
}

int main() {
    fun(3);
    return 0;
}
```

c)

```
#include <stdio.h>

int main(int argc, char *argv[])
{
    int x = 1, z[2] = {10, 11};
    int *p = NULL;

    p = &x;
    *p = 10;

    p = &z[1];

    *(&z[0] + 1) += 3;

    printf("%d, %d, %d\n", x,
z[0], z[1]);

    return 0;
}
```

d)

```
#include <stdio.h>
int fun() {
    static int x = 1;
    return x++;
}

int main() {
    printf("%d ", fun());
    printf("%d ", fun());
    printf("%d ", fun());
    return 0;
}
```

e)

```
#include <stdio.h>

int test(int x)
{
    return x + 1;
}

int main() {
    printf("%d",
test(test(5)));
    return 0;
}
```

f)

```
#include <stdio.h>

int funcp() {
    static int x = 1;
    x++;
    return x;
}

int main() {
    int x, y;

    x = funcp();
    y = funcp() + x;

    printf("%d\n", (x + y));
    return 0;
}
```

g)

```
#include <stdio.h>

int main() {
    int a = 5;
    int b = a++ + ++a;

    printf("%d %d", a, b);
    return 0;
}
```

h)

```
#include <stdio.h>

void fun(int x) {
    x = x + 5;
}

int main() {
    int x = 10;
    fun(x);
    printf("%d", x);
}
```



**Vishwakarma Government  
Engineering College**  
Department of Information Technology

THE  
**TECH TREASURE**

VOLUME 06 '25

**Editorial Team**

**Editorial Team**

Neha Patel

Anand Upasak

Hem Shah

Umesh Pal

Saumil Nainuji

Jaydip Devaliya

Milan Gohil

**Designed By**

Darshna Gangoda

**Guided By**

Prof. Naimisha Trivedi