

Bhavesh Singhal

+91 9518208482 | worknbhavesh@gmail.com | linkedin.com/in/bhavesh-singhal | github.com/bhaveshsinghal95182

SUMMARY

Full-stack developer building web, desktop, and data products across **TypeScript/React**, **Node.js/Go/Python**, and modern data tooling. Ships secure, containerized systems (**Docker/Nginx**) with **REST/GraphQL** APIs, SQL/NoSQL persistence (**PostgreSQL/MongoDB**), and polished UI translated from **Figma** into responsive components.

EDUCATION

Maharishi Markandeshwar (Deemed to be University)

B.Tech in Computer Science and Engineering

Police DAV Public School

Class XII (CBSE) – 81% | Class X (CBSE) – 89.9%

Ambala, Haryana

Aug 2023 – April 2027

Ambala Cantt, Haryana

2020 – 2022

TECHNICAL SKILLS

Languages: TypeScript, JavaScript (ES6+), Python, Go, Rust, SQL, Java, C++, HTML, CSS, Lua

Frontend: React, Next.js, Vite, Electron, Tauri

UI & Styling: Tailwind CSS, shadcn/ui, Radix UI, Framer Motion, GSAP, Monaco Editor

Backend & APIs: Node.js, Express.js, FastAPI, Flask, Go Fiber, Socket.IO, REST APIs, WebSockets, GraphQL, JSON

Data & Storage: PostgreSQL, MongoDB, Redis, Drizzle ORM, Google Cloud Storage

Auth & Security: Clerk, JWT

Media & Video: Remotion, FFmpeg

Local AI: Ollama, Mistral

Data Science: Pandas, NumPy, scikit-learn, TensorFlow, NLTK, Matplotlib, Seaborn

Tools: Git/GitHub, Docker, Docker Compose, Nginx, Linux, pnpm, npm, ESLint, Prettier, Postman, Vercel, Figma, Neovim

EXPERIENCE

Freelance Frontend Developer

Self-Employed

Jun 2024 – Present

Remote

- Delivered **2** production-ready waitlist landing pages (Speechy, PureHydro), generating **500+** email signups within the first month of launch.
- Engineered Upstash Redis rate limiting to sustain **1000+** concurrent requests without degradation or downtime.
- Optimized Core Web Vitals to **98+** Lighthouse by applying lazy loading, code splitting, and next-gen image compression.

Software Engineer Intern

Supertek

May 2025 – Jun 2025

Remote

- Developed production-grade **React + TypeScript** UI, shipping **3** reusable components adopted across the core product within a **6-week** sprint.
- Collaborated with backend engineers to integrate **REST APIs** and **WebSockets**, enabling real-time data sync across **4** dashboard views.
- Resolved critical issues in a legacy **PHP** codebase, reducing the defect backlog by **15%** while improving production stability.

PROJECTS

Code Execution Platform (Live) | *Next.js, TypeScript, Express.js, Docker, Linux, Monaco Editor, Nginx*

- Produced a secure, Dockerized **/api/execute** service supporting **4** languages (JavaScript, Python, C++, Java) and handling **50+** concurrent runs.
- Implemented per-request Linux user isolation with **ulimit** guards (**10s** CPU, **40-process** cap) and process-group cleanup to prevent zombie processes.
- Operated production services via Docker Compose with Nginx TLS reverse proxying, maintaining parity between development and deployment environments.

VideoShare (Secure Video Sharing Platform) (Live) | *Next.js, PostgreSQL, Drizzle, Clerk, Google Cloud Storage, Tailwind*

- Built a Clerk-authenticated Next.js dashboard for secure video uploads, searchable libraries, and client-ready sharing flows.
- Integrated PostgreSQL with Drizzle ORM and Google Cloud Storage streaming to persist metadata and deliver playback without exposing direct object URLs.
- Strengthened content protection with configurable watermarking and time-limited share tokens plus layered download prevention.

Messy-UI (Animated Component Library) (Live) | *React, Next.js, GSAP, Framer Motion, Tailwind CSS, shadcn/ui*

- Designed an animated component registry with reusable composition patterns and production-grade documentation for React teams.
- Standardized a shadcn/ui-compatible distribution workflow by generating registry JSON manifests, reducing adoption to a single CLI command.
- Enabled dual rendering modes (Inline, Sandboxed) to validate components both in isolation and in full-page interactive demos.

Vim Interactive Tutor (Live) | *React 19, TypeScript, Vite, Tailwind CSS, shadcn/ui, Radix UI*

- Constructed a web-based **vimtutor** platform featuring **24** interactive lessons with real-time validation, hints, and progressive unlocks.
- Programmed Vim emulation across Normal/Insert/Visual/Command/Replace modes with robust parsing for command chains like **3w** and **:!ls**.
- Structured a local-first progression system using **localStorage**, enabling instant resume of lessons with zero backend dependency.

Recso (Screen Studio Alternative) | *React, Electron, TypeScript, Remotion, FFmpeg*

- Assembled a cross-platform desktop recorder using React + Electron, capturing desktop streams via Chromium APIs to streamline demo video creation.
- Directed a Remotion + FFmpeg composition pipeline that auto-zooms on user clicks, producing polished exports with only **2** interactions.
- Launched companion landing pages with responsive UI and repeatable release workflows for ongoing product iteration.

Sutra (Obsidian Alternative) | *Rust, TypeScript, Tauri, Vite, React*

- Created a native note-taking app with Tauri + Rust, achieving sub-**100ms** startup and **60%** lower memory usage versus Electron alternatives.
- Instituted a Vite-powered React interface with real-time Markdown rendering and local-first filesystem persistence through Tauri secure IPC.
- Published the project as fully open source while keeping a performance-first architecture to minimize runtime overhead.

TweakCN CSS Import (Figma Community Plugin) (Repo) | *TypeScript, Figma Plugin API, CSS Variables*

- Conceptualized an offline Figma Community plugin that converts TweakCN/custom CSS into a color palette and variable collections for design token handoff.
- Generated palettes by parsing CSS variables locally with zero network access, keeping theming workflows reproducible inside Figma.
- Attained **182** users on the official Community listing and shipped a Version **2** bug-fix release (Aug 2025).

Kickstart Express v2 (Express Scaffolder CLI) (npm) | *TypeScript, Node.js, Express.js, Docker, CLI*

- Authored the **kickstart-express** v2 CLI to scaffold Express servers with TypeScript/JavaScript, modern tooling, and optional Docker setup.
- Introduced an **add** workflow to extend existing codebases with database (MongoDB/PostgreSQL; Mongoose/Prisma/Drizzle) and auth (JWT/Clerk) modules.
- Streamlined backend bootstrapping by generating structured controller/service/route layouts with hot-reload scripts and lint-ready defaults.

Auto-CIDE (Collaborative IDE) (Live) | *Node.js, Socket.IO, Docker, MongoDB, Redis, Monaco Editor, JWT*

- Orchestrated a real-time collaborative IDE with Monaco Editor and Socket.IO, synchronizing multi-user edits across connected clients.
- Unified frontend, backend, and MongoDB services behind Docker Compose for one-command startup and consistent local environments.
- Reinforced session security with JWT authentication and optional Redis caching hooks to support scalable collaboration workloads.

LeetCode API Data Service (Live) | *Python, FastAPI, GraphQL, REST APIs, JSON*

- Collected LeetCode problem metadata via GraphQL on a daily refresh cycle, maintaining an up-to-date dataset for analytics use cases.
- Expanded the public API to **8+** FastAPI endpoints for search, problem lookup, user stats, random picks, and daily challenge retrieval.
- Presented interactive Swagger docs (*/docs*) alongside a Vercel-hosted deployment, enabling frictionless consumption without API keys.

Endpoint Tester MCP Server | *Python, FastMCP, HTTP, JSON*

- Established an MCP server built on FastMCP that exposes **3** reusable HTTP tools for agent-driven API requests.
- Interpreted responses by **Content-Type**, returning JSON as dictionaries, text as strings, and binary payloads as bytes for simpler downstream handling.
- Customized requests with configurable headers, query parameters, bodies, and timeouts to support robust integration testing.

Pollution Flag Analytics Backend | *Python, Pandas, TensorFlow, scikit-learn, Flask*

- Systematized data ingestion and preprocessing for emissions datasets using pandas, producing model-ready features through consistent encoding.
- Trained and evaluated a pollution-flag classifier up to **96%** validation accuracy through iterative tuning and learning-rate scheduling.
- Visualized emission distributions via a Flask CSV-upload endpoint that outputs charts as PNGs, accelerating exploratory analysis.

Fake Review Detection Pipeline | *Python, Pandas, NLTK, scikit-learn, Seaborn*

- Audited **40,432** product reviews with pandas profiling, null checks, and rating distribution analysis to baseline dataset quality.
- Devised a reusable NLP cleaning pipeline with token normalization, stop-word removal, stemming, and lemmatization to improve feature consistency.
- Prepared cleaned corpora for TF-IDF vectorization and classification experiments, speeding iteration on baseline models.

Resume MCP Server | *TypeScript, Node.js, MCP SDK, LaTeX*

- Simplified resume generation by building an MCP server that outputs raw LaTeX from structured projects, skills, and education inputs, reducing manual tailoring time by **90%**.
- Defined tool interfaces (**search-projects**, **get-experience**, **create-resume-latex***) with strict schemas and error handling for reliable automation.
- Minimized agent-to-data latency by using **StdioServerTransport** for local, low-overhead tool execution.

Automations (Local LLM Scheduling Bot) | *Python, Ollama, Mistral, JSON*

- Formulated a Python scheduling bot that queries Mistral via Ollama and answers daily agenda questions from a JSON timetable.
- Organized a modular automation repository to support future Jarvis-style integrations while keeping scripts composable.
- Reduced dependence on external APIs by running inference locally and avoiding API keys for routine planning tasks.

Kahoot Clone (Quiz Backend) | *Go 1.22, Fiber, MongoDB, REST APIs*

- Handled Go Fiber routing for quiz sessions, backing endpoints like */api/quizes* with MongoDB collections for content retrieval.
- Validated request payloads and error paths to keep API responses lightweight and predictable for interactive gameplay.
- Improved service foundations (routing, validation, logging) to align with high-throughput, low-latency game traffic patterns.

CERTIFICATIONS & ACHIEVEMENTS

Oracle Cloud Infrastructure 2025 Certified Developer Professional

Oct 2025

Achieved **1st** place out of **500+** teams and placed Top-10 finalist in **3** competitive hackathons by shipping production-grade applications under time constraints.

Earned Microsoft Partner Center verification and obtained official code-signing certificates for Windows application distribution.

Maintained **90+** GitHub repositories spanning frontend systems, desktop apps, tooling, and full-stack projects.