

Profile Summary

Name: Lakshya Paliwal

Education: Third-year Computer Science student, Manipal University Jaipur

Origin: Gurgaon, India

Primary Interests: Full-stack development, Large Language Models, Generative AI, DevOps, System Design

Professional Focus

- Building scalable full-stack applications
 - Designing AI-powered systems
 - Working with WebRTC, cloud infrastructure, and AI evaluation pipelines
 - Exploring LLM reasoning, temporal understanding, and multi-agent systems
-

Work Experience

Oddmind Innovations — AI Interview Platform

Duration: June 2025 – November 2025

Website: <https://interviews.zariya.ai/>

Context:

Startup focused on building an AI-powered interview preparation platform for consulting, growth marketing, and product management case interviews.

Core Responsibilities:

- Integrated **LiveKit (WebRTC)** for real-time AI-driven interviews
- Migrated away from OpenAI-based conversational flow due to long-context instability
- Implemented **Razorpay** for secure payment processing
- Designed frontend-based interview recording pipeline
- Built **AWS S3 multipart upload pipeline** for video recordings (90% upload time reduction)
- Integrated **Gemini** for interview evaluation and feedback generation
- Implemented **cron-based background jobs** for delayed feedback email delivery (runs every ~30 minutes)

Details about LiveKit, multipart uploads, or Gemini evaluation should only be surfaced when explicitly queried.

Responenet — Full stack & Intern

Duration: December 2025 – Present

Organization Website: <https://www.responenet.org/>

Project 1: ImpactVolunteer

Website: <https://impactvolunteer.com/>

High-level Description:

A multi-role platform connecting volunteers, nonprofits, corporations (CSR), and academic institutions.

Key Contributions:

- Refactored a severely unstructured legacy codebase (\~1000+ files)
- Re-designed UI and core workflows for readability, performance, and maintainability
- Improved architectural consistency across the codebase

Detailed platform role explanations should only be returned if specifically asked.

Project 2: Responenet Website Migration

Objective:

Migrate a 20+ year-old WordPress website to **Next.js** without breaking SEO or existing URLs.

Key Contributions:

- Automated scraping of:
- SEO metadata (OG tags, Twitter cards, canonical URLs)
- JSON-LD structured data
- Downloaded and restructured all WordPress content and media assets
- Preserved URL structure to avoid SEO regressions
- Migrated content into a structured Next.js architecture
- YouTube-to-Website Automation\Automatically sync YouTube uploads to the website and database.

Implementation of the Youtube-to-website feature:

- Used **PubSubHubbub** (publish-subscribe + webhook architecture)
- Subscribed to YouTube channel feeds
- On new video upload:
- Webhook triggers backend ingestion
- Video stored as embed entry in database

Research Experience

Chronocept Project — Temporal Reasoning in LLMs

Paper: <https://arxiv.org/abs/2505.07637>

Research Goal:

Enable large language models to better understand and reason about time.

Role: Data Annotation & Benchmark Design

Key Contributions (High-level):

- Annotated temporal datasets using a structured, multi-step protocol
- Worked on two custom benchmarks for temporal supervision

Annotation Framework:

- **Temporal Segmentation:** Broke text into timeline-consistent segments
- **Primary Axis Classification:** One of eight temporal axes:
 - Main, Intention, Opinion, Hypothetical, Generic, Negation, Static, Recurrent
- **Secondary Axis:** Temporal validity distributions annotated using skewed normal distributions (ξ , ω ,
a) on a logarithmic time scale

Quality Control:

- Dual-annotator system per sample
- Periodic audits of 10% annotations
- Strict agreement thresholds (ICC, Jaccard Index, PrI)
- Ambiguous samples preserved via union-based resolution

Statistical thresholds and annotation mechanics should only be explained when explicitly requested.

Projects

Smart Voice Navigator

Github: <https://github.com/21lakshh/Smart-Voice-Navigator>

Description:

A real-time visual navigation assistant for visually impaired users, built using a **multi-agent state machine** inside the LiveKit ecosystem.

Architecture Overview:

- **Central Reasoning Engine:** Google Gemini Flash
- **Agent Pipeline:**
 - Intent & location capture
 - Object detection using **YOLOv11**
 - Semantic matching using **SentenceTransformers (MiniLM)** for label robustness
 - Depth estimation for spatial awareness
 - Fallback **RAG agent** querying a structured household knowledge base when objects are not detected

Interaction Layer:

- OpenAI STT for speech recognition
 - Sarvam TTS for low-latency voice responses
-

Finfluenzz

Github: <https://github.com/21lakshh/Finfluenzz>

Live: <https://finfluenzz.vercel.app/>

Description:

A retro-gaming-inspired, AI-powered fintech platform designed to improve financial literacy for Gen Z through gamification.

Core Features:

- AI finance advisor (Groq + Gemini)
- Stock & crypto analysis with technical indicators (RSI, SMA, MACD)
- Portfolio risk and diversification analysis
- Real-time financial news aggregation
- Budget tracking with AI-generated spending summaries
- Gamified XP, levels, and achievement system

Tech Stack:

- Frontend: React 18, TypeScript, Vite, Tailwind CSS
 - Backend: Cloudflare Workers, Hono, Prisma, Neon PostgreSQL
 - APIs: Groq, Gemini, Alpha Vantage, Finnhub, CoinGecko
 - Auth & Security: JWT, bcrypt, RBAC
-

Snippify

Github: <https://github.com/21lakshh/Snippify>

Live: <https://snippify-zeta.vercel.app/>

Description:

A full-stack developer platform for discovering, managing, and AI-generating reusable code snippets.

Core Capabilities:

- CRUD operations for snippets (public & private)
- AI-powered code generation, explanation, and refactoring
- Curated UI component library

Tech Stack:

- Frontend: React, TypeScript, Tailwind CSS
 - Backend: Cloudflare Workers, Hono
 - Database: PostgreSQL
 - AI: Groq Cloud LLMs
-

RealM — AI-Powered Information Verifier

Github: <https://github.com/21lakshh/RealM>

Description:

A Chrome extension that verifies online claims and detects misinformation by combining AI-based claim extraction with real-time web search verification.

Core Capabilities:

- Screenshot-based claim capture from any website
- AI-driven core claim extraction
- Real-time fact verification using live web search
- Verdict generation with supporting reasoning and sources

System Design (High-level):

- Claim extraction and reasoning powered by **Google Gemini**
- Web evidence retrieval via **Tavily Search API**
- Stateless verification backend deployed on **Cloudflare Workers**
- Chrome extension frontend for screen capture and user interaction

Tech Stack:

- Backend: Hono, Cloudflare Workers, TypeScript
- AI: Google Gemini
- Search: Tavily
- Extension: React, TypeScript, Tailwind CSS, Vite

Implementation details such as screenshot processing, claim extraction prompts, or verification logic should only be surfaced when explicitly queried.

Additional Links for more Projects

- GitHub: <https://github.com/21lakshh>
-

RAG Usage Notes (Meta)

- Each section is intentionally isolated for selective retrieval
- Avoid returning multi-section responses unless explicitly requested
- Prefer answering with the **smallest relevant subsection**
- Expand only when the user asks follow-up questions