

Pratham Goyal

+91 7489234727 | pratham.goyal2303@gmail.com | [LinkedIn](#) | [GitHub](#) | [LeetCode](#) | [Codeforces](#)

SUMMARY

Software Development Engineer (SDE) Intern candidate graduating in 2027 (B.Tech, ECE) who designs and ships backend systems in Java, Spring Boot, and Python. Strong CS fundamentals – data structures, algorithms, object-oriented design, complexity analysis – with hands-on work in distributed, multi-tiered systems. 700+ problems solved; Codeforces Div. 2 global rank 501.

EDUCATION

Shri G.S. Institute of Technology & Science (SGSITS), Indore, MP

2023 – 2027

B.Tech. in Electronics & Communication Engineering

Relevant Coursework: Data Structures & Algorithms, Operating Systems, DBMS, Computer Networks, Object-Oriented Programming, Computer Architecture.

EXPERIENCE

Freelance Web Developer | SS Skin & Pedia Clinic (**Live Site**) (Client)

2025 – Present

Remote – Indore, India · Tech: React.js, JavaScript, Google Maps API, WhatsApp API, Netlify

- Delivered and maintain a live production website for a two-doctor medical clinic serving 250+ patients – online appointment booking, WhatsApp integration, and Google Maps.
- Worked directly with the clients on requirements and owned end-to-end delivery: responsive React UI (desktop/tablet/mobile), deployment, and ongoing maintenance on Netlify.

PROJECTS

High-Concurrency Distributed Rate Limiter | [GitHub](#)

Aug 2026

Tech Stack: Java, Spring Boot, Redis, Lua, Docker, Docker Compose, Linux, Python

- Built a distributed rate limiter on the token-bucket algorithm with Redis-backed shared state, enforcing consistent request quotas across horizontally scaled, independent server instances.
- Diagnosed a race condition on a non-atomic GET-then-SET path where 30 concurrent requests for one user all passed a 10-token bucket (30 admitted vs. 10 allowed); eliminated it with a single atomic Redis Lua check-and-decrement that caps admissions at exactly 10 under identical concurrent load.
- Wrote a multithreaded Python harness that deterministically reproduces the race and proves correctness after the fix; packaged as a multi-stage Docker build via Docker Compose for one-command reproducible deployment.

Settlement Reconciliation Agent | [GitHub](#)

Aug 2026

Tech Stack: Python, FastAPI, Docker, Gemini API

- Built a finance-ops reconciliation pipeline that auto-matches merchant orders against aggregator settlements, closing 81% of a 63-record batch deterministically with Decimal-precise arithmetic and a centralized tolerance rule shared across layers.
- Bounded an LLM exception-classifier behind a validation guard that re-checks every model claim against the raw settlement data, rejecting arithmetically-invalid “fabrications” – so the model can never move money on numbers it invents.
- Backed by 49 unit tests across matcher, classifier, guard, and API layers, plus a FastAPI /reconcile endpoint and a Dockerized one-command run.

TECHNICAL SKILLS

- **Languages:** Java, C++, C, Python, JavaScript, SQL
- **Backend:** Spring Boot, REST APIs, FastAPI, WebSockets, Microservices, Multithreading, Concurrency
- **Databases & Infra:** MySQL, MongoDB, Redis, Docker, Docker Compose, Maven, Linux, Git
- **Concepts:** Object-Oriented Design, Low-Level Design, Distributed Systems, Multi-tiered Systems, Scalability, Caching, Data Structures & Algorithms, Algorithm Design, Complexity Analysis, Operating Systems, Computer Networks, DBMS, Agile

ACHIEVEMENTS

- Secured **Global Rank 501** out of 28,000+ participants in Codeforces Round 1115 (Div. 2).
- Ranked in the **top 6%** of 15,000+ participants in the Goldman Sachs Hackathon.
- Solved **700+** algorithmic problems across LeetCode and Codeforces.
- Shortlisted for **Amazon ML Summer School 2026** through a competitive resume and assessment screen.