# Rahul Gupta

in [LinkedIn](#) | 🌐 [rahulgpt.com](#) | ✉ rahulr09gupta@gmail.com | ⊙ [GitHub](#)

## Skills

- JavaScript | TypeScript | Python(familiar) | Java | C | PostgresQL | Node | Nestjs | React | Redux | Terraform | MongoDB | Git
- AWS | Cloud Computing | CI/CD | Jest | Cucumber | Docker | Kafka | Unit Testing | Serverless | OOP | Figma
- Microservices | Distributed Systems | Frontend | Backend | Full-Stack | English, Hindi – *All professional proficiency or above*

## Experience

**Fullstack Developer Associate**          **Prominence Games**          *Remote*     **03/2023 - Present**

- Led the implementation of a robust DevOps pipeline, integrating automated testing, continuous integration, and deployment strategies, thereby accelerating the software development lifecycle.
- Led the implementation of Infrastructure as Code (IaC) using **Terraform**, streamlining the deployment process and ensuring consistent environments across development, staging, and production which **r**educed the time it took to manually set up infrastructure using aws console for an environment **by 80%** and **improved scalability** of the system by using **Auto Scaling Groups** (ASG) and **ALB**.
- Integrated wallet signing to generate access tokens and refresh token pairs, enhancing the security of the authentication process.
- Designed and implemented an innovative solution for dynamically generating NFTs by layering various assets to display a placeholder image to end users and asynchronously generating the final NFT, resulting in an **80% reduction in generation pipeline costs and time**.
- Proposed and evaluated potential solutions to reconcile dropped transactions including: Cron job polling to periodically update metadata and user data based on the latest blockchain transactions.

**Software Engineer Intern**          **Prominence Games**          *Remote*     **03/2022 - 08/2023**

- Enhanced security and optimized Axios integration by implementing a robust solution to store authentication tokens (JWT) securely in an HTTP-only signed cookie
- Successfully integrated Storybook into the frontend design system, enabling isolated component testing across various states with a test coverage of 30%.
- Effectively restructured 15% of the database schema to seamlessly accommodate new frontend changes and ensure data consistency.
- Successfully integrated the Ronin (EVM blockchain) API into the Content Management System, streamlining the purchase order flow for improved efficiency and functionality.
- Enhanced the user experience (UX) by refining five UI components through subtle design enhancements and user feedback.
- **Leveraged knowledge in** Git, **Javascript**, **Typescript**, **React**, **Nest**, Umi and REST architecture.

**Software Engineer Intern**          **IkanBilis**          *Remote*     **08/2021 - 10/2021**

- Developed a server-side app and integrated Discord-Js library for Ikan Bilis Community, **boosting engagement by 40%** with moderation, invite stats, XP tracking, and leaderboards.
- Incorporated persistent storage with caching for efficient management of recently used items.
- Executed AB testing, funneled data via a customer data platform (Segment) to Facebook Pixel and Google Analytics, achieving a 65% coverage rate.
- Created a Proof-of-Concept for application load testing using Locust and delivered comprehensive analytics.
- *Leveraged Knowledge in* Git, Python, **Javascript**, **React**, Express, Redis, **MongoDB** and CronTab.

## Education

**Bachelor of Technology**          **Rajasthan Technical University**          *Jaipur, RJ, India*     **08/2019 - 05/2023**

- Major in Computer Science Engineering

## Projects

**KRAKEN** (HTTP Server)**:**
- Developed a C-based HTTP web server from the ground up. [Github](#), [Documentation Site](#) **(03/2023)**
- Crafted the project entirely in C, alongside a dedicated data structure library, Owl ([Github](#)), tailored to its requirements.
- Designed a custom data structure library to encompass essential structures like **Hashmap**, **LinkedList**, **Queue**, and **Threadpool**.
- Implemented and utilized a thread pool to enhance service throughput, enabling concurrent handling of multiple blocking requests.
- The project also features a documentation site, featuring examples and API specifications.

**SORTVIZ** (Sorting Visualizer)
- Created a Sorting Visualizer in **React**. Link to [Github](#), [Demo](#) **(08/2021)**
- The visualizer features 5 **sorting algorithms:** Bubble, Selection, Quick, Heap and Insertion.
- The project includes a built-in drawer for displaying additional algorithm information and supports hotkeys.