Shivam Kumar Singh

+91 9625124611 | shivamsai006@example.com

www.linkedin.com/in/shsaish008| https://github.com/shsaish006

Objective

Motivated and detail-oriented Computer Science student with a strong foundation in programming languages such as C++, Python, and SQL. Passionate about software development, problem-solving, and writing clean, efficient code. Seeking an opportunity to apply my coding skills and logical thinking to real-world challenges, contribute to impactful projects, and continuously learn and grow in a dynamic tech environment.

Experience

- **Programming Languages:** C, C++, Java, Python, SQL, PL/SQL, TypeScript
- **Web Development:** HTML5, CSS3, JavaScript, CSS Flexbox, Bootstrap, React.js, Express.js, Node.js
- **DevOps & Tools:** Git, Docker, Kubernetes, Bash, Linux
- **Cloud Computing:** AWS (Lambda, EC2), Cloud Computing (IaaS)
- **Databases & Backend:** MongoDB, SQL, PL/SQL, Apache Kafka, DBMS
- **Core CS Concepts:** Operating Systems, Data Structures, OOP, Game Theory
- **Other Skills:** Analytical Skills, Microsoft Excel, Engineering Fundamentals
- **Data Science:** NumPy, Pandas, Matplotlib

Education

Lovely Professional University Aug 2023 – Present

B.Tech. in Computer Science and Engineering CGPA: 8.7/10

Relevant Coursework: Object Oriented Programming, Database Management, Discrete Mathematics, Data Structures and Algorithms, Operating Systems, Computer Networks, Cloud Architecture and Infrastructure, Java, Design and Analysis of Algorithms, Computer Organization and Design

Projects

On-Demand Scoring Infrastructure Migration

Tech Stack: AWS Fargate, Lambda, EC2, ECR, Kafka, Parameter Store, TypeScript, Node.js, Java, Docker

- Reduced infrastructure cost by ~80% by replacing always-on EC2 instances with serverless Fargate tasks
- Improved scoring responsiveness by 70%, achieving container startup in under 30 seconds
- Enabled dynamic multi-scorer support and flexible challenge configurations using Parameter Store
- Automated 100% score processing through a Kafka-triggered Lambda pipeline integrated with a review API
- Delivered real-time submission tracking and performance analytics (40% success rate, avg. 3m 37s processing time)
- Increased code reusability by 90% through modularization of scorer logic and challenge-specific separation

Challenge API Dashboard – Full Lifecycle Management for Coding Challenges

Tech Stack: TypeScript, Node.js, Express.js, PostgreSQL, Prisma ORM, REST APIs, HTML/CSS, JavaScript

- Designed and developed a comprehensive Challenge API Dashboard to manage the complete lifecycle of coding challenges—from creation and updates to live status tracking
- Built scalable and type-safe RESTful APIs with full CRUD functionality using PostgreSQL and Prisma ORM for reliable challenge data management
- Implemented a responsive frontend dashboard with ~80% UI coverage, supporting dynamic filters, pagination, and real-time statistics (e.g., Total, Active, Completed challenges)
- Integrated an interactive API Explorer and enriched Swagger documentation, streamlining the developer workflow and reducing API integration effort
- Introduced a Challenge Source Tracking system to categorize and trace challenge origins (e.g., GitHub, Work Manager), improving organizational clarity and internal coordination
- Enhanced user experience by enabling form-based challenge creation and supporting diverse competition types ensuring platform flexibility

KEY PROJECTS

Alternative-Routes in Road Networks
Tech Stack: C++, Graph Theory, Dijkstra's Algorithm, Object-Oriented Programming (OOP), Simulation

- Designed and implemented a simulation of a dynamic road network system using C++, applying Dijkstra's shortest path algorithm to compute the fastest route from source to destination
- Simulated random traffic conditions and congestion across road segments to evaluate route efficiency under varying loads
- Developed logic to prevent vehicle collisions by adjusting vehicle speeds in real-time based on surrounding traffic behavior
- Utilized object-oriented principles to model vehicles, roads, and traffic rules in a modular, maintainable codebase

Lookups API Migration – Modernizing Reference Data Infrastructure
Tech Stack: NestJS, TypeScript, PostgreSQL, Prisma ORM, JWT, DynamoDB, Elasticsearch

- Migrated 100% of the existing API codebase from JavaScript/Express to TypeScript/NestJS, improving type safety and maintainability
- Replaced 2 legacy technologies (DynamoDB and Elasticsearch) with a single PostgreSQL database, reducing infrastructure complexity by ~60%
- Created 3 fully normalized Prisma schema models (Country, Educational Institution, Device) with complete field mapping and relationships
- Developed data migration scripts to transfer over 100,000 records from DynamoDB to PostgreSQL with zero data loss
- Preserved 100% API compatibility, including JWT auth, admin/M2M authorization, query filters, and pagination support
- Reduced local development setup time by ~70% by removing Elasticsearch and DynamoDB dependencies
- Delivered full API documentation (30+ endpoints) using OpenAI and created a Postman test suite for validation
- Wrote 20+ unit tests and seed scripts for critical modules to ensure reliability and enable test automation
- Documented the entire migration and deployment process in a detailed README with step-by-step setup and environment instructions

EXTRACURRICULAR_

- Expert on **Codeforces** with a rating of **1648**, actively solving algorithmic problems on a global platform
- ★★★ (3-Star) Coder on **CodeChef**, consistently participating in rated contests
- ★★★ (3-Star) on **GeeksforGeeks**, solving structured DSA problems with consistency.
- Achieved a rating of **1800+** on **LeetCode**, ranking in the top **9.4% globally** among all users.
- Regular participant in national and global coding contests, with a strong focus on **problem-solving** .