SOURADEEP GHOSH

9606596308 | souradeepghosh608@gmail.com | linkedin

Technical Skills

- **Programming:**Python,C/C++, Java, JavaScript, SQL
- **Web Frameworks:** FastAPI, Flask, Streamlit, Dash
- **Cloud & Deployment:** AWS (Lambda, ECS, S3), GCP, Docker, Kubernetes, CI/CD
- **Databases & Search:** PostgreSQL, MongoDB , NoSQL
- **Data Streaming & Messaging:** Kafka, Redis, Celery
- **Machine Learning/AI:** Scikit-learn, XGBoost, TensorFlow, PyTorch

Relevant coursework

- Data Structures & Algorithms, Operating Systems, Database Management, Software Engineering
- Computer Networks, Distributed Systems , Image processing

Competitive Programming & Achievements

- **CodingNinjas METRON : Rank 15**
- **Codeforces: (ghshsour)** Rated ~**1400(specialist)**, and solved ~**350** problems
- **Codeforces Round 1029 (Div. 3) :** Rank **3462** out of **40,000**+ particiapants
- **Codeforces Round 1034 (Div. 3) :** Rank **4411** out of **40,000**+ particiapants
- **LeetCode Contests:(sour806) Debuted** with **1651** rating; solved **450**+ problems.

Relevant Projects

Patient Management System using Microservices | *Java, Spring Boot, Docker, PostgreSQL,AWS,Kubernetes*

- Faced integration issues while scaling services—RabbitMQ was dropping ~20% of messages during high load.
- Resolved it by implementing message retries, persistent queues, and consumer acknowledgements—cut message loss to 0%.
- Containerized 5+ services via Docker Compose and improved startup efficiency by ~35% using parallel dependency bootstrapping.
- Built a fault-tolerant, event-driven architecture handling 1,000+ simulated user interactions per test cycle.

C++ Compiler | *C++, Flex, Bison, LLVM*

- Designed and implemented a mini compiler for a subset of the C language, covering lexical analysis, syntax analysis, semantic checks, and code generation.
- Built the lexical analyzer using regular expressions and finite automata (DFA/NFA) concepts to tokenize source code.
- Implemented a parser using CFGs, FIRST/FOLLOW sets, and LL(1)/LR parsing techniques, ensuring error handling for invalid syntax.
- Generated intermediate representation (IR) and converted it into simple assembly-like instructions (with LLVM backend integration for optimization and executable code generation).
- Applied automata theory (DFA, CFG, PDA) and compiler design principles learned in TOC and systems courses, bridging theory with real-world implementation.

CodeHermes GenAI Agent for Code Intelligence | *Python, FastAPI, Next.js, Docker, LLM, CI/CD*

- Built a full-stack code intelligence web app using FastAPI (Python) backend & Next.js (TypeScript) frontend.
- Implemented semantic code search and call graph generation for Python & JavaScript codebases, improving search accuracy by 30%.
- Integrated LLM-based QA to answer developer queries, handling high concurrent requests with low latency.
- Deployed on Docker containers with automated CI/CD pipeline.

Education

Kalinga Institute of Industrial Technology **Aug 2023 – July 2027**

Bachelor of Technology in Computer Science and Engineering

CGPA 8.8 *Bhubaneshwar , Odisha*