

# L ASHWINI BALACHANDRAN

+91 9667712326 New Delhi, India [lashwinib@gmail.com](mailto:lashwinib@gmail.com)

[linkedin.com/in/ashwini-lb](https://linkedin.com/in/ashwini-lb) [chickenleaf.onrender.com](https://chickenleaf.onrender.com) [github.com/chickenleaf](https://github.com/chickenleaf) [leetcode.com/u/chickenleaf](https://leetcode.com/u/chickenleaf)

Software Engineer with 1 year of experience, seeking full-time SDE-1 roles.

## EDUCATION

<b>MS Management</b> , Jawaharlal Nehru University, New Delhi	2024
<b>Bachelor of Technology, Computer Science</b> , Jawaharlal Nehru University, New Delhi	2024

## EXPERIENCE

<b>LLM Engineer Intern</b> Sekurennet	Feb 2024 - Jul 2024 <i>Remote</i>
--	--------------------------------------

- Reduced query processing time by 15% using Redis caching and FAISS while building Q&A agents and RAG bots with open-source LLMs (Mistral, Mixtral, LiteLLM, Llama), Qdrant, sentence transformers, and Hugging Face NLP libraries.
- Built & integrated AI assistants for email and calendar tools using APIs and python libraries, improving user productivity by 10% with Phidata, VapiAI, LangChain, OpenAI, Ollama, and Groq Client.
- Developed ClickUp RESTful API tools and PythonREPL script, enhancing testing efficiency for the team.
- Explored self-operating computers with LLMs, YOLO, LLaVA, Moondream, SAM, and PyAutoGUI.

<b>SDE Intern (Python)</b> ThirdOrigin	Jan 2023 - Jun 2023 <i>Remote</i>
---	--------------------------------------

- Reduced processing time by 85.7% by developing core asynchronous function that triggers to further filter and load timezone-based MongoDB data onto the application server.
- Cut data loading time from 14 minutes to under 2 minutes using asynchronous programming, Celery with Redis, and multithreading in batch sizes of 10,000+ optimized via CPU time and memory profiling.
- Developed secure CRUD operations for RDBMS (MySQL, PostgreSQL) and RESTful APIs for NoSQL time-series databases (QuestDB, InfluxDB), using Pydantic & ORM to prevent SQL injection and reduce input errors by 40%.
- Optimized libraries using profiling, batch processing, data structures, and rigorous testing, significantly boosting data handling and system performance by atleast 20%.
- Utilized Sphinx, Jira/Atlassian, Amazon Workspace, Apache Guacamole, Docker, Linux CLI.

## SKILLS

<b>Languages</b>	Python, C/C++
<b>Databases</b>	RDBMS (PostgreSQL, MySQL), NoSQL (MongoDB, Redis, InfluxDB, QuestDB), RabbitMQ
<b>Tools</b>	Django, Flask, FastAPI, Docker, Kubernetes, Celery, Git, Postman, CI/CD, Nginx, ElasticStack
<b>ML/AI</b>	scikit-learn, GridSearchCV, Hugging Face Transformers, LangChain, OpenCV, YOLO, SAM

## PROJECTS

**Portfolio Website:** Bootstrap, Django, SQLite, Render, Docker, CI/CD, Webhook, Gunicorn, Whitenoise, Networking.

**QueryJet API:** FastAPI, Elasticsearch, Celery, RabbitMQ, Docker, Kubernetes. Can process over 50,000 requests per second with sub-second query response times.

**WheresMyStuff? API:** [WIP] FastAPI, MongoDB, Elasticsearch, Celery, RabbitMQ, Logstash, Kibana, Docker.

**Kaggle Titanic- ML Notebook:** Pandas, Matplotlib, scikit-learn, Support Vector Classifier, Random Forest, GridSearchCV. Ranked in the top 1% on the leaderboard.

**NativeFlow:** Transformers Library, MarianMT, Langdetect, NLP, FastAPI, SQLAlchemy, PostgreSQL, Docker.