

# Rajeev Atla

US Citizen | [732-209-3995](tel:732-209-3995) | [rajeevatla101@gmail.com](mailto:rajeevatla101@gmail.com) | [github.com/RajeevAtla](https://github.com/RajeevAtla) | [linkedin.com/in/rajeev-atla](https://linkedin.com/in/rajeev-atla) | [rajeevatla.com](http://rajeevatla.com)

## EDUCATION

<b>Rutgers University - School of Engineering</b> <i>Master of Science in Computer Engineering (Specialization in Machine Learning)</i>	Sep 2025 — May 2026 New Brunswick, NJ
Coursework: Reinforcement Learning, Multimodal AI, High Performance/Distributed Computing	

  

<b>Rutgers University - School of Engineering</b> <i>Bachelor of Science (Triple Major) in Computer Engineering, Computer Science, and Data Science</i>	Sep 2021 — May 2025 New Brunswick, NJ
Recipient of the Eleanor and Samuel Sneath Endowed Merit Scholarship for Engineering Students	

Coursework: AI, Distributed Deep Learning, Data Science, Statistical Learning, Computer Vision

## SKILLS

- **Programming Languages:** Python, R, SQL, Java, C/C++/CUDA, Rust, Bash
- **AI/ML Libraries:** NumPy, PyTorch, JAX, TensorFlow, Keras, Pandas, Scikit-Learn, NLTK, LangChain/LangGraph
- **Data Visualization:** Matplotlib, Seaborn, Plotly, Tableau
- **Cloud & DevOps:** AWS, Microsoft Azure, OCI (Oracle Cloud Infrastructure), GitHub Actions, Docker, Kubernetes
- **Tools & Databases:** Jupyter Notebooks, Apache Kafka, Git, Linux (Ubuntu), PostgreSQL, MongoDB, Jira

## CERTIFICATIONS

- AWS: [Certified Cloud Practitioner](#), [Certified Machine Learning Specialist](#), [Certified AI Practitioner](#)
- Oracle (OCI): [AI Foundations Associate](#), [Generative AI Professional](#), [Data Science Professional](#), [Vector AI Search Professional](#)

## WORK EXPERIENCE

<b>Software Engineering Intern</b> Atlait Inc.	May 2024 — September 2024 Remote
• Developed a Python-SQL compression script for form data, <b>reducing storage costs by 7%</b> for enterprise clients	

• Integrated PyTorch inference into Kafka-microservices architecture, **improving mean response time by 96 milliseconds**

• Updated codebase from ES5 to ES7 using HTML, CSS, and TypeScript, resulting in **23% faster mean page loads**

• Optimized CI/CD pipeline to **speed up build times by 13%** ensuring efficient development cycles

## PROJECTS

<b>dexMCP</b>	<a href="https://bit.ly/dexmcp">https://bit.ly/dexmcp</a>
• Engineered Model Context Protocol (MCP) server exposing <b>5+ reusable tools</b> and <b>5+ Pydantic models</b>	
• Implemented parameter validation across <b>20+ typed fields</b> and <b>100% of tool inputs</b>	
• Built asynchronous clients using DSPy and LangChain to auto-discover tools and execute multi-step requests	
<b>SuperconGAN</b>	<a href="https://bit.ly/3z7JaqZ">https://bit.ly/3z7JaqZ</a>
• Built a PyTorch-based GAN to model superconductivity data of various materials, enhancing generative AI applications	
• Extracted and processed <b>80,000+ dataset entries</b> from the UCI ML Repository using Pandas efficiently	
• Released Python package on PyPI, achieving over <b>80,000+ downloads</b> and widespread adoption	
• Authored a LaTeX paper on findings and future scope, <b>incorporating 500,000+ data points</b> effectively	
<b>Cityscape (2nd Overall at HackExeter 2021)</b>	<a href="https://bit.ly/3OZjJ07">https://bit.ly/3OZjJ07</a>
• Led a <b>team of 4</b> in designing and implementing a city tour mobile app, resulting in <b>100+ vivid city tours</b> for users	
• Wrote controllers and models for MongoDB using MongooseORM to store <b>30+ kB of geographic data</b> in NoSQL schema	
• Built mobile user interface allowing users to search, review, rank, and explore <b>100+ tours</b> using Flutter/Dart	
• Constructed REST API using Express.js and nodemon to <b>increase development velocity by 20%</b> with hot-reloading	
<b>DocuMint</b>	<a href="https://bit.ly/DocuMint">https://bit.ly/DocuMint</a>
• Built a 5-agent LangGraph + Gemini doc-modernizer with Gradio, achieved <b>90%+ modernization coverage</b> on sample docs, <b>cut manual edit time 50%</b> with a <b>4-tab UX</b> , hardened with <b>8 deterministic pytest cases</b> and network-safe skips	
• Authored modular agents (fetcher, analyzer, researcher, generator, quality-checker) with structured prompts and severity-prioritized research, <b>lifting modernization accuracy by 35%</b> and <b>trimming LLM API spend by 20%</b> through top-issue capping, content truncation, and batching	