HARSHIT JAIN

+1 (582) 203-9755 ◇ harshitj.cs@gmail.com ◇ linkedin.com/in/harshitjain17 ◇ github.com/harshitjain17

EDUCATION

The Pennsylvania State University, University Park PA Expected Graduation: Dec 2024
Bachelor of Science in Computer Science
GPA: 3.7/4.0 | Dean's List (5/5) | Technical Coach @ CodePath | AlgoPSU Captain @ ACM | Resident Assistant
Relevant Coursework: Data Structures and Algorithms, Math of Machine Learning, Operating Systems Design, Systems Programming, Supervised Machine Learning, Advanced Learning Algorithms (Deep Learning), Generative AI with LLMs, Theory of Computation, CodePath: (Intermediate+Advanced) Software Engineering, Database Management Systems

TECHNICAL SKILLS

Programming Languages: Python, C/C++, JavaScript, Java, HTML/CSS, MATLAB, Verilog, Assembly (64/32-bit x86)
Frameworks & Tools: TensorFlow/Keras, Scikit-Learn, Numpy, Node.js, React.js, Next.js, RESTful APIs, LaTeX, Git
Softwares: AWS, GCP, LLMs, MS SQL Server/MySQL/PostgreSQL, Linux/UNIX, SonarQube, Postman, Bitbucket, JIRA

WORK EXPERIENCE

Software Engineer Intern May 2024 - Present
Hughes Network Systems, LLC Germantown, MD

- Working within their Aviation business (Aero program)

Machine Learning Engineer Intern Jan 2024 - May 2024
Materials Research Institute (2DCC-MIP Team), Penn State University University Park, PA

- Leveraged 3 SOTA LLMs (GPT-4, Jurassic2, and Llama2) for chatbot-integrated answer retrieval using the RAG Model
- Architected a Next.js app with end-to-end AWS integration, optimizing RESTful APIs for scalability and deployment
- Automated Python-based AWS Lambdas for video processing and transcription, handling 50% user upload surge
- Implemented AWS Rekognition for face recognition, enabling automatic tagging and improving content organization accuracy
- Engineered optimal AWS EC2 instance selection, reducing launch times by 25% through precise instance matching
- Utilized: Python, AWS, Next.js, LLMs, Deep Learning Models

Software Engineer Co-op May 2023 - Dec 2023
VIAVI Solutions Inc. Germantown, MD

- Collaborated with the 6-person R&D team to architect and implement a Python-based automated test suite on Linux systems for PNT instruments, ensuring exhaustive test coverage and strict compliance with the SCPI protocol
- Debugged PNT unit source code in C/C++, leading to a 55% reduction in bugs and a 30% increase in code coverage
- Executed 35+ long-term tests on core devices using SCPI commands, ensuring adherence to release-level quality standards
- Utilized: Python, C/C++, SCPI Protocol, Bitbucket, Confluence, SonarQube, Git, Agile, JIRA

Software Engineer Intern May 2022 - May 2023
Materials Research Institute (2DCC-MIP Team), Penn State University University Park, PA

- Engineered a React.js app, designing 50+ user-facing features across 20+ REACT components (utilized Jest testing)
- Integrated an MS SQL Server relational database to manage data from 500+ instruments across 18+ tables
- Automated Python scripts for data retrieval, manipulation, and integration with robust error handling and efficient filtering
- Research: Developed a Python library for the Raman Fitting model, enabling deconvolution of Raman spectra. The library facilitated interactive preprocessing, effective fitting, and data export, reducing analysis time by 40% (tested)
- Utilized: Python, JavaScript, React.js, Node.js, MS SQL Server, RESTful APIs, Git, HTML/CSS, JIRA

PROJECTS

Dynamic Memory Allocator [C/C++] Jan 2024 - Feb 2024

- Designed custom malloc, free, realloc; segregated free lists and footer optimization to improve memory management
- Achieved a utilization score of 69% and benchmark throughput at 100% across diverse computing environments

HiLite: AI AutoHighlighter [Python, Flask, React.js] Mar 2023 - May 2023

- Designed an AI system that summarizes text using Long Short-Term Memory (LSTM) networks
- Trained the model on the training set, using the validation set to monitor its performance and prevent overfitting

mdadm Linear Device [C/C++, Linux] Feb 2023 - May 2023

- Configured 16 disks of size 64 KB as a 1MB linear device, providing users with a unified address space for data access
- Implemented mount/unmount operations to the linear device, mitigating potential data loss and system crashes
- Designed the read/write functions to set up in the linear device, providing users with comprehensive data access capabilities
- Engineered data caching solution to enhance system latency reduced I/O wait time by 60%