Sammy Dabbas

Orlando, FL — (407) 800-2029 — dabbassammy@gmail.com — LinkedIn — GitHub

Education

University of Central Florida **Expected Dec 2026**
B.S. in Computer Science **GPA: 3.8**

Skills

Languages: Python, C#, Java, JavaScript, C++, SQL
Frameworks/Libraries: FastAPI, Django, React.js, .NET Core, Pandas, NumPy
Databases: Amazon DynamoDB, MS SQL Server, Firebase Firestore
Cloud: AWS (Kinesis, DynamoDB, DAX, S3, EKS)
Tools & Ops: Git, GitHub, Docker, Kubernetes, Jenkins, CloudWatch, X Ray, Visual Studio, FFmpeg
Testing: unit, integration, basic load testing; code reviews

Work Experience

FINDER Software Solutions Orlando, FL
Software Developer Intern *Sep 2024 to Present*

- Shipped Python and C# services in production. Distributed RTSP streaming across many servers ingests **millions of vehicle events per week**; node alerts; deployments managed by **Jenkins**.
- Built ETL pipelines processing **10+ GB/day**; improved ingestion accuracy by 15% and reduced latency by 25%.
- Created high throughput .NET parsers validating multi GB SQL/XML datasets; reduced processing errors by **50%**.
- Added unit and integration tests and basic load checks as gates; participated in reviews on every change.

Research Experience

NASA Research Team Member — *Python, Pandas, Matplotlib*

- Forecasted severe weather and performed error analysis, improving launch prediction accuracy by 5% for Kennedy Space Center.
- Analyzed 10+ years of weather fatality data and surfaced insights adopted by the National Weather Service.
- Built a Python tool estimating Lightning Launch Commit Criteria violations to support operations.

Farm Social Media Engagement Research — *Python, Pandas, Scikit learn, TensorFlow*

- Built pipelines to scrape and transform posts from 100+ farm pages; normalized text, timing, and image metadata into a clean dataset.
- Trained and evaluated models to predict engagement and compute feature importance; iterated on features and validation.

Projects

Real Time Wiki Change Processing System — *Python, FastAPI, AWS (Kinesis, DynamoDB, DAX)*

- EventStreams → Kinesis → consumers → DynamoDB; FastAPI control and query endpoints; Firehose archives raw events to S3.
- Revision ID dedup and hot key cache keep reads under <**50 ms**; autoscaling across stream, compute, and DB; metrics and traces in CloudWatch and X Ray; unit and integration tests.

GPU Training Service (team of 3) — *Django, React, Docker, Kubernetes*

- Full stack platform to deploy and manage ML workloads, cutting setup time by **40%**; simulated a cluster of **16** GPU nodes.
- Designed Django APIs and containerized training environments for scalable orchestration.