

VISHNU VIVEK VALETI

Gainesville, FL | +1 (352) 226-7455 | vishnuvivek.valeti@gmail.com | linkedin.com/in/valetivishnuvivek | github.com/valetivivek

EDUCATION

University of Florida

Aug 2024 – May 2026

Master of Science, Computer Science and Engineering

GPA: 3.83

Relevant Coursework: Distributed Operating Systems, Software Engineering, Advanced Data Structures, NLP, Analysis of Algorithms, Computer Networks

Vignan University

Aug 2020 – May 2024

Bachelor of Technology, Information Technology

GPA: 3.43

Relevant Coursework: OOP Through Java, Python Programming, Database Management Systems, Computer Organization & Architecture, IoT, Networks & Computer Security, Operating Systems

TECHNICAL SKILLS

Languages: Python, Go, JavaScript, TypeScript, Java, C/C++, SQL, Bash

Frameworks: React.js, Node.js, Express.js, FastAPI, Celery, PyTorch, TensorFlow

Databases: PostgreSQL, MySQL, MongoDB, Redis, Cassandra, DynamoDB.

Cloud & DevOps: AWS (EC2, S3, Lambda, EKS, SageMaker), Docker, Kubernetes, Terraform, GitHub Actions

Tools & Methodologies: Git, REST APIs, gRPC, System Design, Microservices, Agile/Scrum, Jest, Prometheus, Grafana

EXPERIENCE

Software Engineer Intern - ML

Dec 2025 – Present

ReplyQuick LLC

USA (Remote)

- Designed scalable ML training pipelines for the DentalScan platform, processing **5,000+ dental images** with structured preprocessing, **augmentation**, and validation to support disease classification and region-level detection
- Developed evaluation utilities in **PyTorch** to compute and **log classification + detection metrics** (precision/recall, confusion matrix, **IoU-based scoring**), reducing experiment comparison time by **30%**
- Integrated ML inference outputs with backend APIs by defining **versioned response schemas** and consistent payload formats, enabling automated result delivery and reducing manual post-processing
- Implemented model versioning and deployment support by packaging inference artifacts with structured logging and config-based runs, improving reproducibility across retraining cycles

Graduate Student Assistant

Aug 2025 – Dec 2025

University of Florida

Gainesville, FL

- Designed and maintained ML training-data pipeline using **AWS S3, EC2, and PostgreSQL**, managing structured datasets and improving data retrieval efficiency by **35%** for model training workflows
- Developed backend APIs using **Python (FastAPI)** and containerized with **Docker**, reducing result-serving latency from **900ms to 280ms** via query optimization and indexing
- Integrated backend APIs with the frontend dashboard to display casino fairness rankings, reducing analysis time by **30%**

Software Development Intern

Jan 2024 – May 2024

Vignan University

Guntur, India

- Contributed to feature enhancements and bug fixes for a **React/MySQL** academic portal used by **2,500+ users daily**, improving reliability across attendance and staff workflows
- Implemented attendance automation features (**mark-all** and **quick-edit**) to reduce repetitive per-student actions, saving **~8 hours/week per department** across 5 - 6 departments
- Improved results-page performance by identifying slow **JOIN-heavy** queries and optimizing with **MySQL indexing** and query refactoring, reducing load time from **3.2s to 1.0s**
- Strengthened security for grade/results pages by implementing **RBAC** and **OAuth2.0**, preventing unauthorized access and improving account safety

TECHNICAL PROJECTS

Webhook Dispatcher

Python, FastAPI, Redis, Celery, PostgreSQL, Docker, Prometheus

- Built a Dockerized webhook dispatcher in **Python + FastAPI**, enqueueing events in **Redis** and processing deliveries asynchronously via **Celery** workers
- Implemented at-least-once delivery with backoff retries and **idempotency keys** to avoid duplicate sends reliably end-to-end
- Modeled delivery state in **PostgreSQL** and exposed **Prometheus** metrics (success rate, queue depth, p95 latency) to monitor throughput and reliability

Reddit Clone

Go, HTTP Server, Actor Model, Token Bucket, CLI

- Built an actor-based social media platform in **Go** using message-passing to model users, subreddits, posts, and comments
- Managed concurrent in-memory state with **sync.Map** and **RWMutex** to ensure thread-safe operations across users
- Implemented token-bucket rate limiting using **Go** at middleware and handler levels to control request throughput