James Galbreath

CS-499

8/3/25

Part 1

Trend 1: Edge Computing

Edge computing moves data processing from centralized servers to devices near the data source, like IoT sensors and smart devices. This reduces latency, saves bandwidth, and supports real-time decision-making for applications such as autonomous vehicles, industrial automation, and smart cities.

It drives the need for distributed architectures and software that works in resource‑constrained, variable‑network environments. This increases demand for skills in distributed algorithms, microservices at the edge, and hybrid edge‑cloud systems.  
Consumers gain faster, more responsive services; workers in manufacturing, logistics, and healthcare benefit from real-time analytics; and citizens see smarter infrastructure. This aligns with my goal of developing scalable, high‑performance distributed systems.

Trend 2: Federated Learning

Federated learning trains AI models locally on devices and shares only model updates, keeping raw data private. This improves privacy, regulatory compliance, and efficiency, especially in sensitive fields like healthcare, finance, and mobile computing.  
It advances privacy‑preserving AI by requiring algorithms that handle uneven data, unreliable connectivity, and secure aggregation. This complements edge computing by enabling AI training where data is collected.

Consumers enjoy smarter apps without losing privacy; workers in regulated industries gain safe AI tools; and citizens benefit from ethical AI in public services. This matches my interest in secure, distributed computing and building privacy‑first AI systems.

So far, I have achieved several course outcomes, including designing and implementing complex software solutions (SceneManager.cpp), applying and optimizing algorithms in the DAD Final Project, and developing relational databases with CRUD functionality (AnimalShelter.py). I have also applied secure coding practices, such as SEI CERT C++ standards and SQL injection prevention, and effectively communicated technical work through documentation and presentations. Remaining outcomes include applying distributed systems and privacy‑preserving AI concepts, such as edge computing and federated learning, in practical environments and further optimizing algorithms for large‑scale real‑time processing.

Part 2

For the databases category, I used my AnimalShelter.py project as the artifact. This project demonstrates my ability to design and implement a relational database, perform CRUD operations, and ensure data integrity. It also integrates Python with SQLite to manage shelter data efficiently. I have applied secure coding practices by validating input and preventing SQL injection vulnerabilities. My progress in this category shows that I can build, query, and maintain databases while following best practices for security and performance.

|  |  |  |  |
| --- | --- | --- | --- |
| Checkpoint | Software Design and Engineering | Algorithms and Data Structures | Databases |
| Name of Artifact Used | SceneManager.cpp | DAD Final Project | AnimalShelter.py |
| Status of Initial Enhancement | DONE | DONE | DONE |
| Submission Status | DONE | DONE | DONE |
| Status of Final Enhancement | DONE | DONE | NO |
| Uploaded to ePortfolio | NO | NO | NO |
| Status of Finalized ePortfolio | NO | NO | NO |