**Artifact 1: CS 330 Final Project — 3D Desk Scene in C++ with OpenGL**

**Project Description:**  
 In this project, I developed a 3D desk scene application using C++ and the OpenGL graphics library within the Visual Studio environment. The core objective was to demonstrate proficiency in object-oriented programming by designing modular components such as SceneManager and ShaderManager. The scene featured multiple 3D objects with basic lighting and texture mapping applied to simulate realistic surfaces and depth.

**Enhancements Implemented:**  
To significantly improve the visual quality and user experience, I planned and executed several enhancements:

* **Advanced Lighting for Complex Objects:** I implemented sophisticated lighting models including multiple light sources and dynamic shadows. This required understanding and coding shaders that calculate ambient, diffuse, and specular lighting, which added realism to objects such as lamps, books, and desktop items.
* **Texturing Objects in the Scene:** I applied detailed texture mapping using image files for surfaces like wood, metal, and fabric, enhancing visual fidelity. This involved learning texture coordinate mapping and blending techniques to avoid distortion or repetition.
* **Interactivity in the 3D Scene:** I introduced interactive controls, allowing users to manipulate the camera angle, zoom, and object positions via keyboard shortcuts and mouse input. This increased the application’s usability and showcased event-driven programming skills.

**Skills Demonstrated:**

* Mastery of advanced C++ features and OpenGL graphics pipeline
* Shader programming and GPU-based lighting calculations
* Integration of texture mapping and interactive input handling
* Application of object-oriented design patterns to maintain clean and modular code

**Program Outcomes:**  
 This project highlights my ability to design complex software systems that integrate graphical rendering and user interaction (Outcome 1 & 2). Additionally, the communication of technical concepts through clear code structure and documentation demonstrates Outcome 5.

**Artifact 2: CS 340 Database Projects**

**Project Description:**  
 This two-phase project began with designing and implementing a MongoDB database system to manage data for an animal shelter. Using Python, I built backend CRUD (Create, Read, Update, Delete) routines to enable robust data manipulation and ensure database integrity.

The second phase involved creating a fully functional, client-facing MongoDB dashboard for Grazioso Salvare, the project’s client. This dashboard was designed using the Dash and Flask frameworks to provide a responsive and interactive interface for shelter administrators to visualize and manage animal data efficiently.

**Enhancements Implemented:**

* **MongoDB CRUD Operations:** Established reliable data management through Python scripts, handling error cases and ensuring atomic operations. This formed the essential backbone for data consistency.
* **Real-Time Dashboard Development:** Leveraged MongoDB change streams to implement real-time updates on the dashboard, allowing immediate reflection of database changes without manual refresh.
* **Advanced Filtering and Visualization:** Integrated dynamic filters for species, age, and status, accompanied by visual charts and tables for insightful reporting.
* **Data Export Features:** Added functionality to export filtered data into CSV and PDF formats, facilitating external reporting and analysis.
* **REST API Design:** Developed a Flask-based RESTful API that decoupled frontend and backend, supporting scalability and future integrations.

**Skills Demonstrated:**

* Full-stack development combining backend database operations with frontend UI/UX design
* Real-time data streaming and asynchronous programming concepts
* RESTful API architecture and data serialization
* User-centric design thinking focused on client needs and usability

**Program Outcomes:**  
 This artifact exhibits proficiency in system design and implementation (Outcome 2), use of modern tools and technologies (Outcome 4), and professional communication through client-oriented solutions (Outcome 5). Problem-solving skills in handling real-time data and integration showcase Outcome 1.

**Artifact 3: CS 305 Secure File Transmission**

**Project Description:**  
This project aimed to create a secure file transmission system that protects data integrity and confidentiality during transfer. The system was built using Spring Boot with a focus on applying current encryption standards and secure communication protocols.

**Enhancements Implemented:**

* **Implementation of Modern Encryption Algorithms:** Integrated SHA-256, SHA-512, and BLAKE2 hashing algorithms to ensure robust verification of transmitted data. Benchmarking these algorithms for speed and security helped in selecting the optimal option dynamically.
* **Secure Communication Channels:** Configured HTTPS with SSL/TLS protocols to encrypt data in transit, preventing eavesdropping and man-in-the-middle attacks.
* **Security Testing Compliance:** Developed and incorporated automated testing aligned with OWASP standards to detect vulnerabilities such as injection attacks, improper authentication, and data exposure.
* **User Feedback on Security Performance:** Added features that provide users with real-time performance metrics and security status, improving transparency and trust.

**Skills Demonstrated:**

* Deep understanding of cryptographic principles and secure protocol implementation
* Application of software security testing methodologies and tools
* Development of secure, maintainable, and testable code
* Communicating security features effectively to end users

**Program Outcomes:**  
 This artifact reflects my capability to apply security principles in software development (Outcome 3), problem-solving in designing secure systems (Outcome 1), and leveraging modern development tools and practices (Outcome 4).