### **📄 Artifact Narrative – Company Offices App (Databases / Full-Stack)**

**Brief Description:** The Company Offices App is a full-stack web application built using a React frontend and a MongoDB backend. It presents company data by founding year and by state using charts and tables for visual clarity and analysis.

**Why I Selected It:** Originally, this artifact was intended to build on code I had written for interacting with a database to display company data. However, I realized that the original project was stored on a remote server I no longer had access to. Rather than abandon the idea, I chose to fully rebuild the entire application from scratch. This required creating a new MongoDB database, designing a backend API to connect to it, and developing a new React-based frontend to present the data.

**Enhancements and Skills Demonstrated:** The enhancements involved a complete rebuild and full implementation of both backend and frontend components. I designed and built a RESTful API to communicate with a MongoDB database, structured and seeded company data, and developed a React interface with dynamic charts and tables. I emphasized a clean UI and made sure the app followed logical component structure and routing.

This project demonstrates my skills in:

* MongoDB schema design and database integration
* Backend development using Node.js and Express
* API creation and data flow management
* Frontend development with React and data visualization tools
* Applying full-stack practices and UI/UX principles

**Course Outcomes Met:**

* **Demonstrate an ability to use innovative tools and frameworks in computing practices**
* **Design and evaluate computing solutions using computer science standards**
* **Communicate technical content through visual and interactive tools**
* **Develop a security mindset (input validation and secure data handling)**

### **📄 Artifact Narrative – Treasure Hunt Game (Software Design & AI Enhancement)**

**Brief Description:** The Treasure Hunt Game is a Python-based program that generates a random maze environment for an AI agent to navigate. The AI must find the optimal path to a goal (the "treasure") through trial and error. The project simulates the application of reinforcement learning principles to solve a dynamic and unpredictable problem space.

**Why I Selected It:** I chose this artifact because it was one of the most intellectually engaging and technically demanding projects I worked on during the program. It not only challenged my understanding of programming and pathfinding but gave me an opportunity to work with early AI concepts. For the capstone, I saw an opportunity to significantly improve the AI’s efficiency by adding a guidance mechanism.

**Enhancements and Skills Demonstrated:** For the enhancement, I developed a new algorithm designed to guide the AI’s learning process more effectively. The original AI solved the maze using basic learning techniques, but the enhanced version included a custom method that offered smarter directional choices during the learning phase. This reduced the number of random, inefficient moves and improved the agent’s overall performance.

This enhancement involved:

* AI agent behavior tuning
* Algorithm design focused on reinforcement learning efficiency
* Software modularization and separation of responsibilities
* Testing and validating improved learning performance

**Course Outcomes Met:**

* **Design and evaluate computing solutions using algorithmic principles**
* **Apply innovative problem-solving techniques in software design**
* **Demonstrate an ability to structure code for flexibility and clarity**
* **Develop a security mindset (ensuring safe navigation and predictable inputs)**

### **📄 Artifact Narrative – Histogram Project (Algorithms and Data Structures)**

**Brief Description:** The Histogram Project is a Python application that reads a text file, analyzes the frequency of items (such as words or letters), and displays the results both in the command-line interface and through a graphical interface using Tkinter. It demonstrates the use of core data structures and algorithmic thinking in a practical text analysis task.

**Why I Selected It:** I selected this project because it allowed me to focus on data structure efficiency and algorithm design. It also gave me a chance to demonstrate how logic developed in a CLI environment could be adapted to a GUI-based interface. The project was a good representation of my ability to take a simple concept—frequency counting—and implement it with attention to both functionality and user experience.

**Enhancements and Skills Demonstrated:** For the capstone, I improved the performance of the frequency analysis by optimizing the core algorithm and separating the logic from the presentation layers. I also extended the program to include a Tkinter GUI that visually displays the histogram, making the program more accessible and visually informative.

Key skills demonstrated include:

* Use of dictionaries and loops for efficient data processing
* Algorithm development and optimization for text parsing
* GUI development using Python's Tkinter library
* Clean code structure with separation of concerns
* Applying object-oriented principles to improve maintainability

**Course Outcomes Met:**

* **Design and evaluate algorithmic solutions to real-world problems**
* **Demonstrate effective use of data structures in software applications**
* **Develop clean, modular, and maintainable code**
* **Deliver technically sound, user-facing tools with visual feedback**