**Narrative for the Weekly Pay Calculator Application**

**Artifact Description:** The Weekly Pay Calculator is a software application designed to compute and store weekly earnings based on daily hours worked and miles driven. Created in 2024, this application utilizes C++ for backend processing, including data storage and user input handling. The application is designed to run in a console environment, providing a simple and straightforward user interface for data entry and retrieval.

**Justification for Inclusion:** This artifact was selected for inclusion in my ePortfolio to demonstrate my abilities in software development, particularly in the areas of algorithm design, data structures, input validation, and file I/O operations. The Weekly Pay Calculator showcases:

* My proficiency in using standard data structures such as std::map and arrays to effectively organize and manage data.
* My ability to implement robust input validation techniques to ensure data integrity.
* My skills in implementing file input/output operations to achieve data persistence, allowing data to be saved across sessions.
* The enhancements made to the application, such as adding persistent data storage and improving user interaction through a menu-driven interface, highlight my ability to iteratively improve and refine software solutions based on user requirements and feedback.

**Reflection on the Enhancement Process:** Enhancing the Weekly Pay Calculator was an enlightening and challenging experience. Through this process, I learned the importance of:

* **User-Centered Design:** Initially, the application only handled session-based data. Feedback from test users highlighted the need for data persistence, leading to the implementation of file storage. This change significantly improved the usability and functionality of the application.
* **Robust Error Handling:** Early versions of the application faced issues with incorrect data entries (e.g., alphabetic characters in numeric fields). Enhancing the input validation logic to handle such cases taught me advanced techniques in error handling and user input processing.
* **Efficiency in Data Management:** Using std::map for storing weekly data allowed for efficient data retrieval and updates, which was crucial for performance as the data volume grew. This taught me to consider the scalability of data structures in software design.

**Challenges Encountered:** Several challenges were encountered and overcome during the enhancement process:

* **File I/O Integration:** Initially, integrating file input/output operations posed challenges, particularly in handling strings with spaces. Adjusting the approach to encapsulate week labels within quotes during file operations resolved this.
* **Input Validation:** Enhancing the input validation to exclude mixed character inputs (e.g., alphanumeric inputs where only numbers were expected) required implementing more complex string parsing strategies.

**Incorporation of Feedback:** Feedback played a crucial role throughout the development process. User feedback was critical in identifying the need for persistent data storage and refining the user interface. Peer reviews helped identify edge cases in input validation that were previously overlooked.

**Artifact Improvement:** The enhancements made to the Weekly Pay Calculator significantly improved its functionality and user experience. The addition of persistent data storage transformed the application from a session-based calculator to a more robust tool capable of handling longitudinal data. Improved input validation mechanisms ensured the application's reliability and robustness.

**Course Outcomes:** This project helped me meet several course outcomes:

* **Software Development Fundamentals:** Demonstrated through the implementation of data structures and basic software logic.
* **Advanced Programming Techniques:** Shown in the handling of file I/O and complex user input validation.
* **Problem Solving and Debugging:** Evident from overcoming challenges related to data persistence and user input errors.

**Unmet Outcomes:** While the project was successful in meeting many outcomes, the graphical user interface (GUI) development aspect was not addressed, which could have further showcased my skills in creating more user-friendly applications.

This narrative underscores not only the technical skills employed during the project but also the iterative learning and problem-solving approach that were fundamental to enhancing the application.