Music Library Project

**Project Description:** Create a Music Library Management System using Python that allows users to organize and search through their music collection efficiently. The system will incorporate search functionality using linear search and implement sorting algorithms (bubble sort, insertion sort, merge sort) to order the music library based on user preferences.

**Skills/Objectives:**

1. **Python Programming:**
   * Utilize Python for implementing the music library management system.
   * Implement algorithms for linear search, bubble sort, insertion sort, and merge sort.
2. **Data Structures:**
   * Represent the music library using appropriate data structures (e.g., arrays, linked lists) to facilitate efficient searching and sorting.
3. **User Interface (UI):**
   * Develop a simple UI using Python libraries like tkinter for user interaction.
   * Enable users to add, remove, and search for songs in the music library.
4. **Algorithmic Understanding:**
   * Demonstrate understanding and application of linear search for finding specific songs.
   * Implement and showcase the functionality of bubble sort, insertion sort, and merge sort for ordering the music library.
5. **Troubleshooting:**
   * Implement error handling mechanisms for potential issues during user interactions, such as adding or removing songs.
   * Troubleshoot and resolve any bugs or errors identified during the testing phase.

**Project Phases:**

1. **Planning:**
   * Define project goals, emphasizing the integration of search and sorting algorithms in the music library system.
   * Identify necessary data structures for efficient implementation.
   * Plan the user interface and user interactions for adding, removing, and searching for songs.
   * Outline the testing strategy, including edge cases and performance testing.
2. **Design:**
   * Develop the basic structure of the music library system, incorporating data structures and algorithms.
   * Implement the UI using tkinter, allowing users to interact with the system.
   * Conduct initial tests to ensure basic functionality.
3. **Testing and Feedback:**
   * Collect bug reports and issues identified during testing.
   * Gather user feedback on the UI and overall user experience.
   * Revise the code and UI based on testing and feedback.
4. **Documentation:**
   * Provide comprehensive setup instructions for running the music library system.
   * Include schematics or flowcharts illustrating the data flow and algorithmic processes.
   * Document the Python code with clear explanations and comments.
   * Develop troubleshooting guidelines for common issues that may arise.

**Python Libraries:**

* Utilize tkinter for creating the graphical user interface.
* Implement algorithms using Python standard libraries.
* Pygame for audio files
* Optionally, use Matplotlib for visualizing the sorting process.