**Guide to Run the Library Management System**

**Overview**

This project is a **Library Management System** developed using **C++** and **wxWidgets**. The system is designed to allow librarians to manage books by providing features for adding, removing, issuing, and returning books. The application uses the **Tree Data Structure** and saves data persistently in a text file (books.txt).

**Prerequisites**

To run this project, ensure the following requirements are met:

**Software Requirements**

1. **C++ Compiler**: Ensure a C++ compiler (e.g., GCC, MSVC) is installed.
2. **wxWidgets Library**: Install the wxWidgets GUI library.
3. **Text Editor/IDE**: Use any C++ compatible editor or IDE (e.g., Visual Studio, Code::Blocks, or VS Code).

**Hardware Requirements**

1. A system capable of running C++ applications.

**Setup Instructions**

**Step 1: Clone the Repository**

1. Open a terminal or command prompt.
2. Clone the GitHub repository using the following command:
3. git clone <repository-link>
4. Navigate to the project directory:
5. cd <repository-folder>

**Step 2: Install wxWidgets**

1. Follow the official wxWidgets installation guide for your operating system:
   * [Windows](https://docs.wxwidgets.org/latest/overview_install.html#overview_install_windows)
   * [Linux](https://docs.wxwidgets.org/latest/overview_install.html#overview_install_unix)
   * [MacOS](https://docs.wxwidgets.org/latest/overview_install.html#overview_install_mac)
2. Ensure wxWidgets is properly linked to your compiler.

**Step 3: Build the Project**

1. Compile the source code by linking the wxWidgets library.
   * Example for GCC:
   * g++ -o LibraryManagement main.cpp `wx-config --cxxflags --libs`
   * For IDEs, configure the project settings to include wxWidgets.
2. Ensure no errors occur during compilation.

**Run Instructions**

1. After a successful build, execute the generated binary:
2. ./LibraryManagement
3. The application window will launch, displaying the Library Management System interface.

**Features and Usage**

**Main Features**

* **Add Book**: Add a new book by entering its ID, title, and author.
* **Remove Book**: Remove an existing book by entering its ID.
* **Issue Book**: Issue a book by entering its ID.
* **Return Book**: Return a previously issued book by entering its ID.
* **Show Available Only**: Toggle to view only the books that are not issued.

**Data Persistence**

* All book data is saved to a text file (books.txt) in the root directory.
* The application automatically loads data from this file upon startup.

**Troubleshooting**

**Common Issues**

1. **Compilation Errors**: Ensure wxWidgets is properly installed and linked to the compiler.
2. **Runtime Errors**: Verify that the books.txt file is accessible and formatted correctly.
3. **GUI Issues**: Check for wxWidgets compatibility with your system.

**Debugging Tips**

* Use logging or breakpoints to track the flow of execution.
* Verify the syntax and paths for wxWidgets in the compilation command.

**Future Enhancements**

* Add search functionality to locate books by title or author.
* Enhance the GUI design for better user experience.
* Implement user authentication for secure librarian access.
* Add support for exporting and importing book data in other formats (e.g., CSV).

**License**

This project is licensed under the MIT License. Feel free to use, modify, and distribute it.

**Contribution**

We welcome contributions to enhance this project. Please follow these steps:

1. Fork the repository.
2. Create a new branch for your feature.
3. Commit and push your changes.
4. Submit a pull request.

Happy Coding! 🎉