Foundation of computer programming
Tutored project
Secure File Synchronization Tool with Version Management using C++ and Qt

Objective: The objective of this project is to develop a comprehensive file synchronization tool with version management capabilities. Students will use C++ as the primary programming language, the Qt framework for the graphical user interface (GUI), and basic socket programming for remote file synchronization. Additionally, the project will involve implementing secure file transfer protocols and utilizing multithreading for enhanced performance.

→Steps:

1. **Project Setup:**
    - Set up a new C++ project using an Integrated Development Environment (IDE) of choice, ensuring the installation of the Qt framework.
    - Configure the project to include necessary Qt modules for GUI development.
2. **Design GUI:**
    - Design a user-friendly GUI using Qt widgets to facilitate file selection, synchronization options, version control, and secure login.
    - Implement a clean and intuitive layout to enhance user experience.
3. **Basic File Synchronization:**
    - Develop the core functionality for local file synchronization, allowing users to select folders, compare file versions, and update files between a local source and destination.
4. **Version Management:**
    - Implement version control for files, keeping track of changes, timestamps, and user modifications.
    - Allow users to roll back to previous versions of files.
5. **Secure Authentication:**
    - Integrate a secure login system to ensure that only authorized users can access the synchronization tool.
    - Implement secure password hashing and storage practices.
6. **Remote File Synchronization:**
    - Extend the tool to support remote file synchronization over a network.
    - Utilize basic socket programming to establish connections between the local and remote instances of the tool.
7. **Multithreading:**
    - Introduce multithreading to enhance the performance of file synchronization processes.
    - Use threads to handle simultaneous file transfers, preventing GUI freezes during large file operations.
8. **Error Handling and Logging:**
    - Implement error handling mechanisms to deal with network issues, file conflicts, and other potential problems.
    - Integrate a logging system to keep track of synchronization activities and errors.
9. **Security Enhancements:**

- Implement secure file transfer protocols, such as SSH or SSL, to encrypt data during remote synchronization.
- Apply best practices for securing file storage and transmission.

10. **User Documentation:**
    - Create comprehensive documentation explaining how to use the tool, including its features, security measures, and troubleshooting steps.
11. **Testing:**
    - Conduct thorough testing to ensure the tool's functionality, security, and performance under various scenarios.
    - Address any bugs or issues identified during testing.
12. **Presentation and Demonstration:**
    - Prepare a presentation showcasing the features, architecture, and implementation details of the file synchronization tool.
    - Demonstrate the tool's usage, emphasizing its user-friendly interface and security measures.
13. **Submission:**
    - Have students submit their source code, documentation, and any additional materials created during the project.

Grading Criteria:

- Code Quality: adherence to C++ best practices, readability, and modularity.
- GUI Design: usability, responsiveness, and adherence to design principles.
- Functionality: completeness of file synchronization features, version control, and secure authentication.
- Network Implementation: successful integration of basic sockets for remote synchronization.
- Multithreading: effective use of threads for improved performance.
- Security Measures: implementation of secure authentication and file transfer protocols.
- Documentation: clarity and completeness of user and technical documentation.

The deadline to return the detailed report and the code : Friday 29th of December 2023.
A presentation is required for each group and you're required to explain your code(random parts)