Feature 1: File Uploading

- The file uploading feature allows users to select and upload files from their local system to the application.
- It provides a user-friendly interface with an "Upload" button, which triggers a file selection dialog when clicked.
- The selected file is then processed using JavaScript's FileReader API to extract relevant information such as the file name, path, and type.
- This information is stored in an array of file objects and displayed in the file list for easy reference.
- The uploaded file can be previewed by generating a URL using the URL.createObjectURL() method.
- This feature enhances the application's functionality by enabling users to conveniently upload files and manage them within the application.

Feature 2: File Locking

- The file-locking feature adds an extra layer of security to the uploaded files.
- Each file in the file list is associated with a lock icon, which indicates its lock status.
- If a file is locked, the lock icon is visible permanently, while if it is unlocked, the lock icon appears only when the file is hovered over.
- Clicking on the lock icon triggers the display of the Lock component, which allows users to set a password for the file or enter the password to open it.
- The Lock component provides two buttons: "Open" to access the file with the correct password and "Unlock" to remove the password from the file.
- This feature provides users with control over file access and ensures the confidentiality of sensitive files.
- It adds an extra layer of security to protect valuable information within the application.

Code Implementation:

- The file uploading functionality is implemented using HTML's input type="file" and JavaScript's FileReader API.
- When a file is selected, the handleFileUpload() function is triggered, which reads the file using FileReader and extracts relevant information.
- The extracted file details are then stored in the application's state using the setMyFiles() function.
- The Lock component is conditionally rendered based on the lock status of each file.
- Clicking on the lock icon or selecting a file triggers the PassManager() function, which handles the display of the Lock component and manages file locking and unlocking based on user actions.
- The Lock component handles password input, locking, unlocking, and validation based on user interactions.

Rationale:

- The file-uploading feature is crucial for allowing users to import files from their local system, making the application a more versatile tool for file management and collaboration.
- The file-locking feature adds an essential layer of security, ensuring that sensitive files are protected within the application.
- By implementing these features, the application becomes a comprehensive file management solution, combining convenience, functionality, and security.
- The file-locking feature enhances data privacy and prevents unauthorized access, which is especially important for confidential or sensitive files.

    - These features were chosen based on their practicality, usability, and the added value they bring to the application, ultimately enhancing the user experience and providing a more robust file management solution.

★ Alongside the mentioned features, I've included a "Refresh" button that restores the application to its default state. This button offers a quick way to reset the interface, removing any temporary changes or data. By clicking "Refresh," users can easily revert back to the original setup, ensuring a clean and seamless experience. This feature enhances usability and provides a straightforward method to refresh the interface for continued productivity.