PDF-Coalesce

INTRODUCTION

The PDF coalesce tool is a versatile solution that allows for various actions on PDF files such as merging, splitting, adding, extracting, and viewing. As a widely-used file format for document sharing across different platforms and devices, PDFs can benefit from the tool's capabilities in managing documents efficiently. Tasks such as combining multiple PDF files into a single document, extracting specific pages, or compressing PDFs can be easily accomplished with this tool, making it a valuable asset for managing PDF files according to your specific needs.

PROBLEM STATEMENT

Working with PDF documents can pose several challenges for users, such as the complexity of combining multiple files into a single document, breaking down large files into smaller ones, inserting additional content into existing files, extracting specific pages from PDF documents, or viewing these files. Furthermore, online PDF tools may be limited by file size or require payment, which further emphasizes the need for a flexible and accessible tool for working with PDF documents.

SOLUTION

The PDF Coalesce is a versatile solution that can address common challenges when working with PDF files, including merging, splitting, adding, extracting, viewing, and annotating. It provides a comprehensive set of features for managing PDF files and is essential for users who regularly work with these documents. Additionally, since the tool is offline, users do not need to upload files, which ensure privacy and confidentiality, particularly for sensitive data.

TECHNOLOGIES/LIBRARIES/MODULES USED

1. **JAVA**-Java is a programming language used for developing desktop and mobile applications.

2. **AWT-**AWT (Abstract Window Toolkit) is a Java GUI library used for creating GUI-based desktop applications.

3. **Swing**-Swing is a Java GUI library that provides a rich set of components for building GUI-based desktop applications.

4. **iTextPDF**-iTextPDF is a library used for creating and manipulating PDF files in Java.

5. **Apache PDFBox**-Apache PDFBox is a Java library used for extracting and splitting PDF files.

The code snippet provided in the question uses Java, AWT, Swing, iTextPDF, and Apache PDFBox libraries.

IMPLEMENTATION

Synergizing Class Concepts with Project

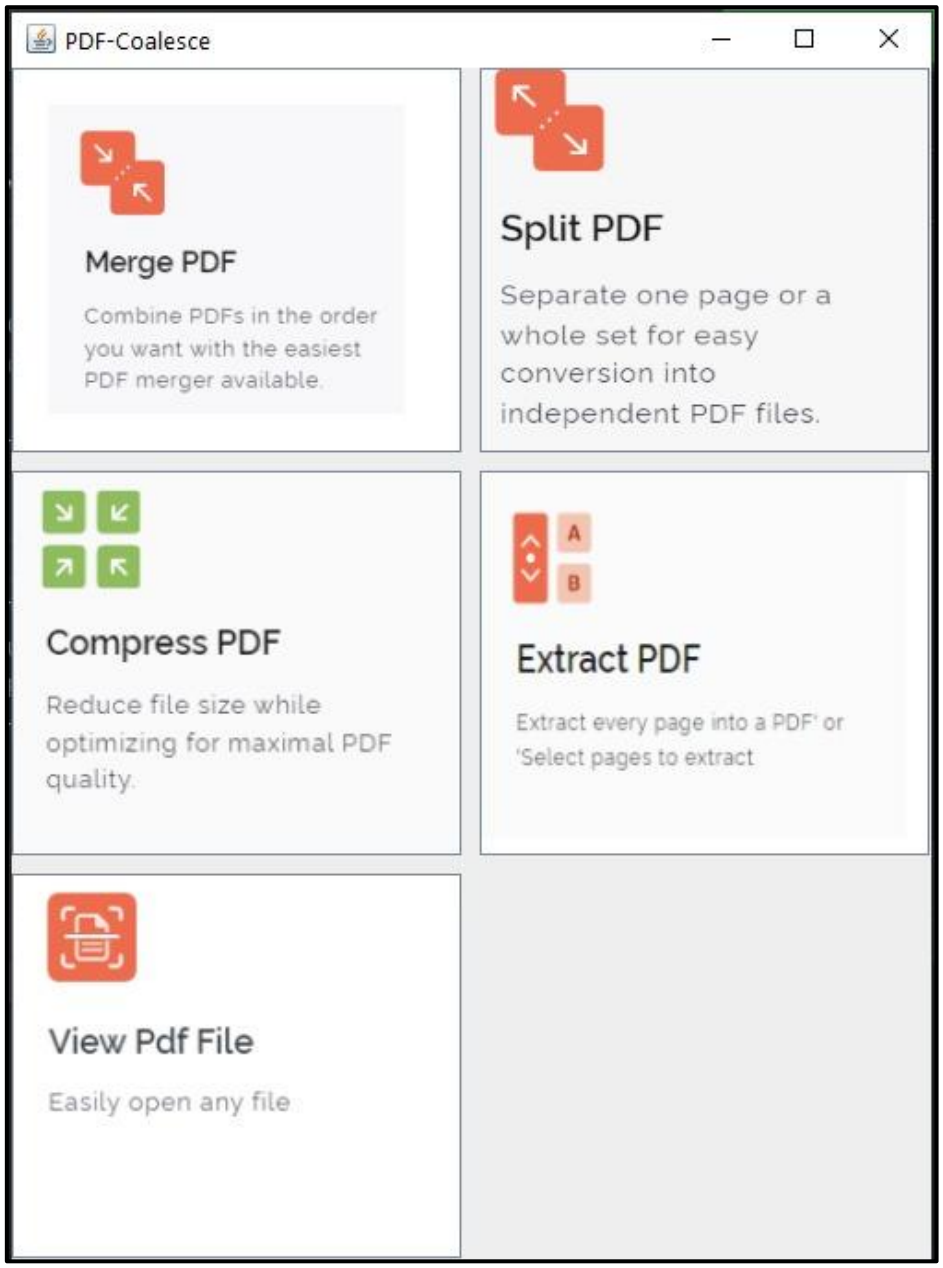
1. Functions and classes
2. Concepts of exception handling
3. Threads implementation
4. Swing
5. AWT
6. Used concept of Importing packages, modules and libraries like pdfbox, itextpdf.
7. Maven concept – POM.XML

RESPONIISBILTY OF EACH GROUP MEMBER

1. Anupam Mittal - Implement Merge and PDF viewer
2. Chandresh Kumar- Compress PDF and GUI of Landing Page
3. Jasmeen Kaur – GUI of functions + Split function
4. Swati Aggarwal – Split + Extract function

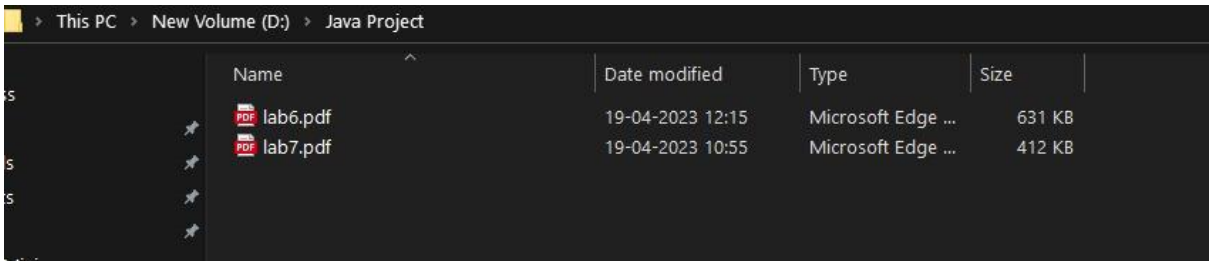
WORKFLOW ALONG WITH SCREENSHOT

1. Landing Page

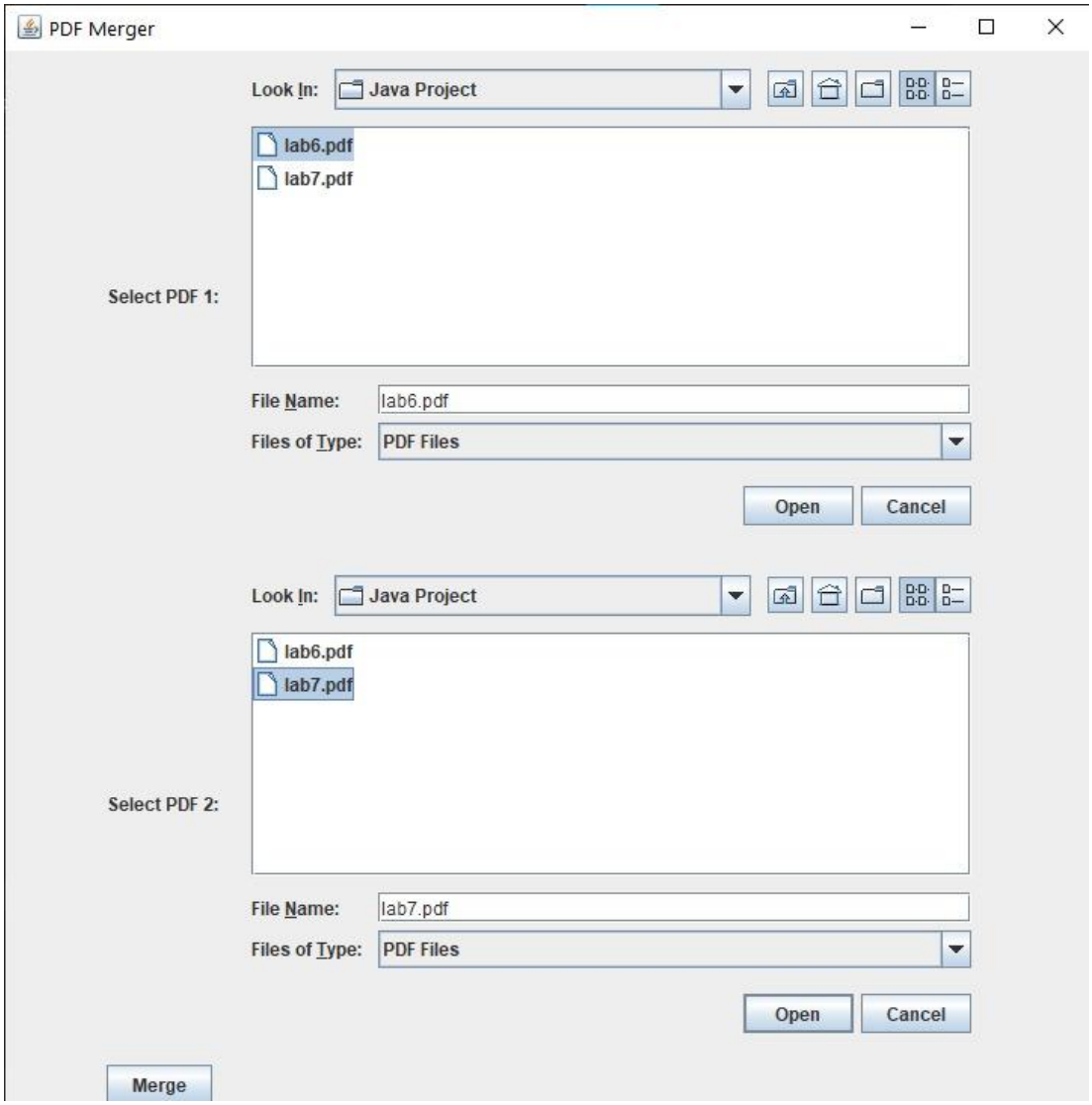


2. Merge Function

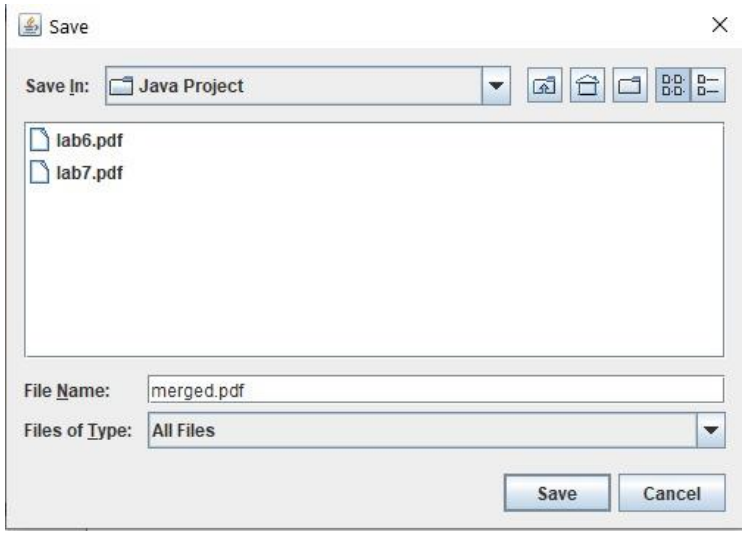
File in The folder

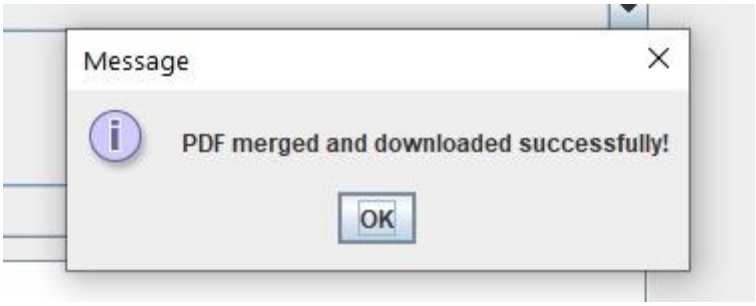


Selecting the File to Merge

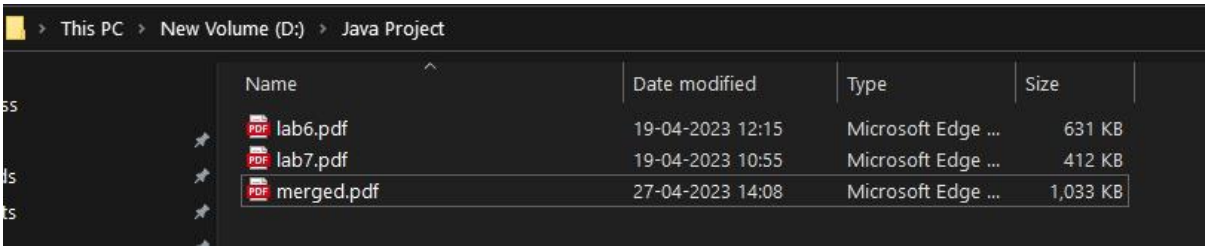


Saving the merged file to the selected location



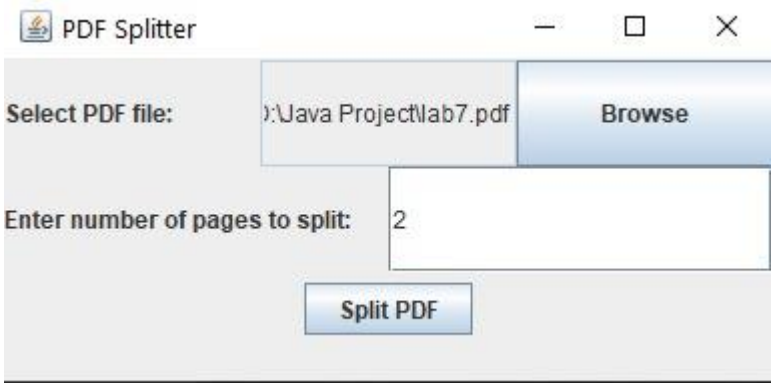


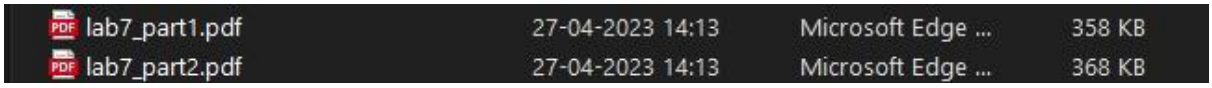
Merged file successfully saved at the selected location



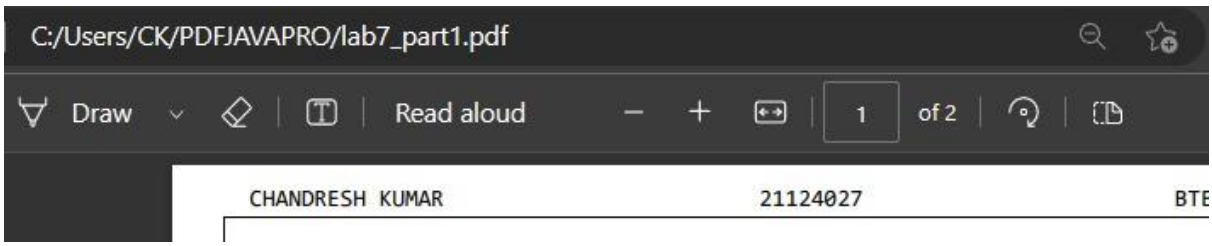
3. Split Function

Selecting the file to split and entering number of pages to split in

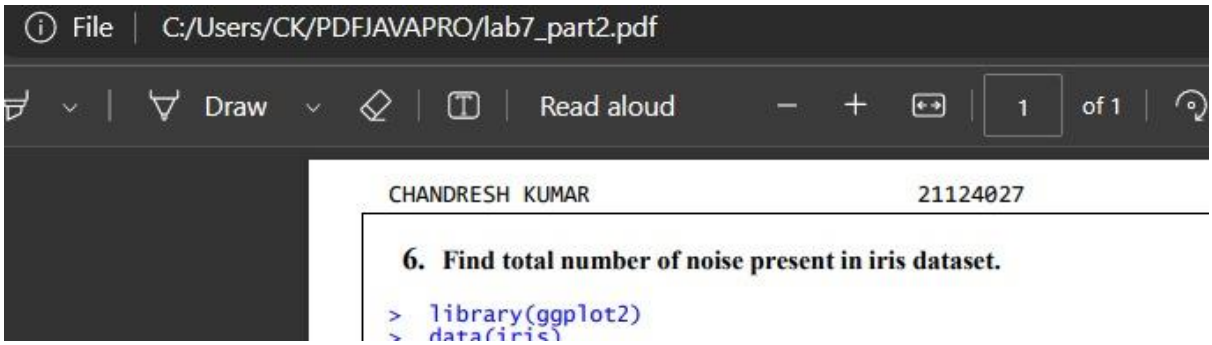




Part 1 of split file

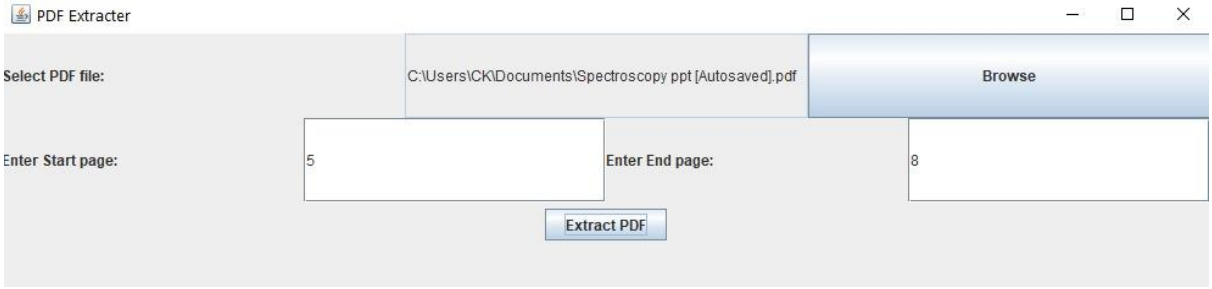


Part 2 of split file

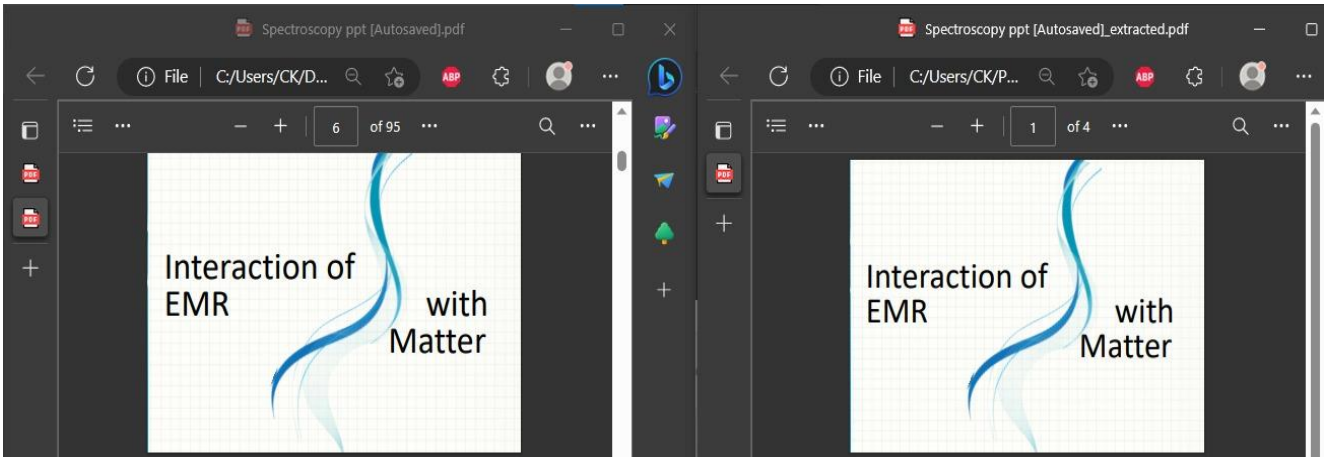


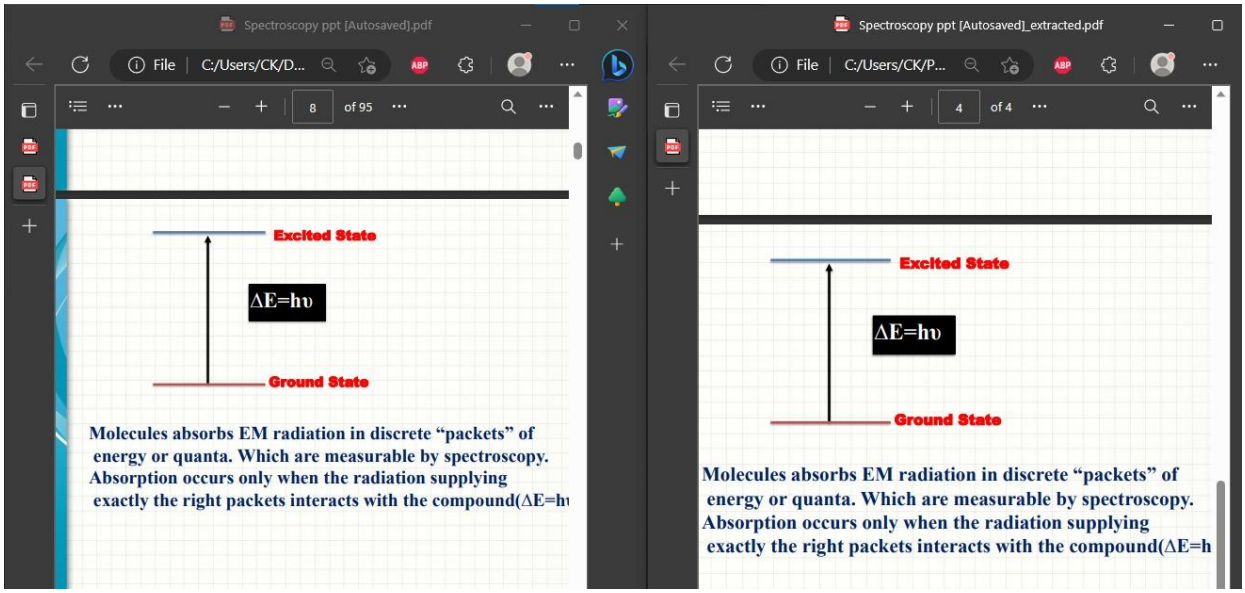
4. Extract File

Selecting the file



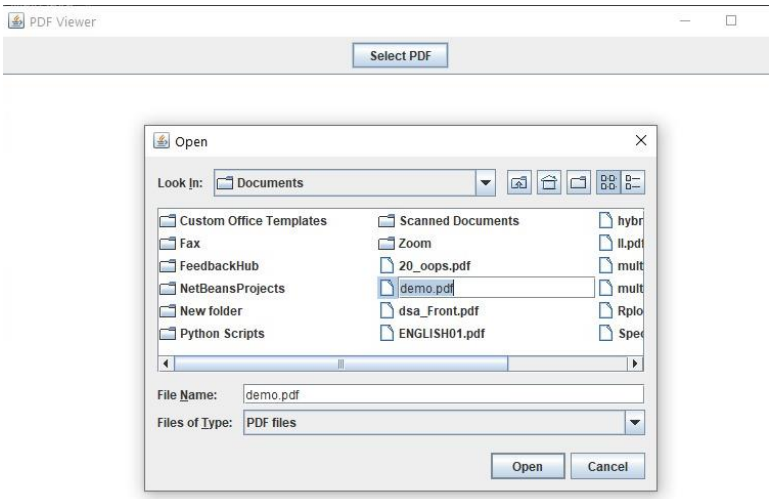
Extracted 4 pages from the file

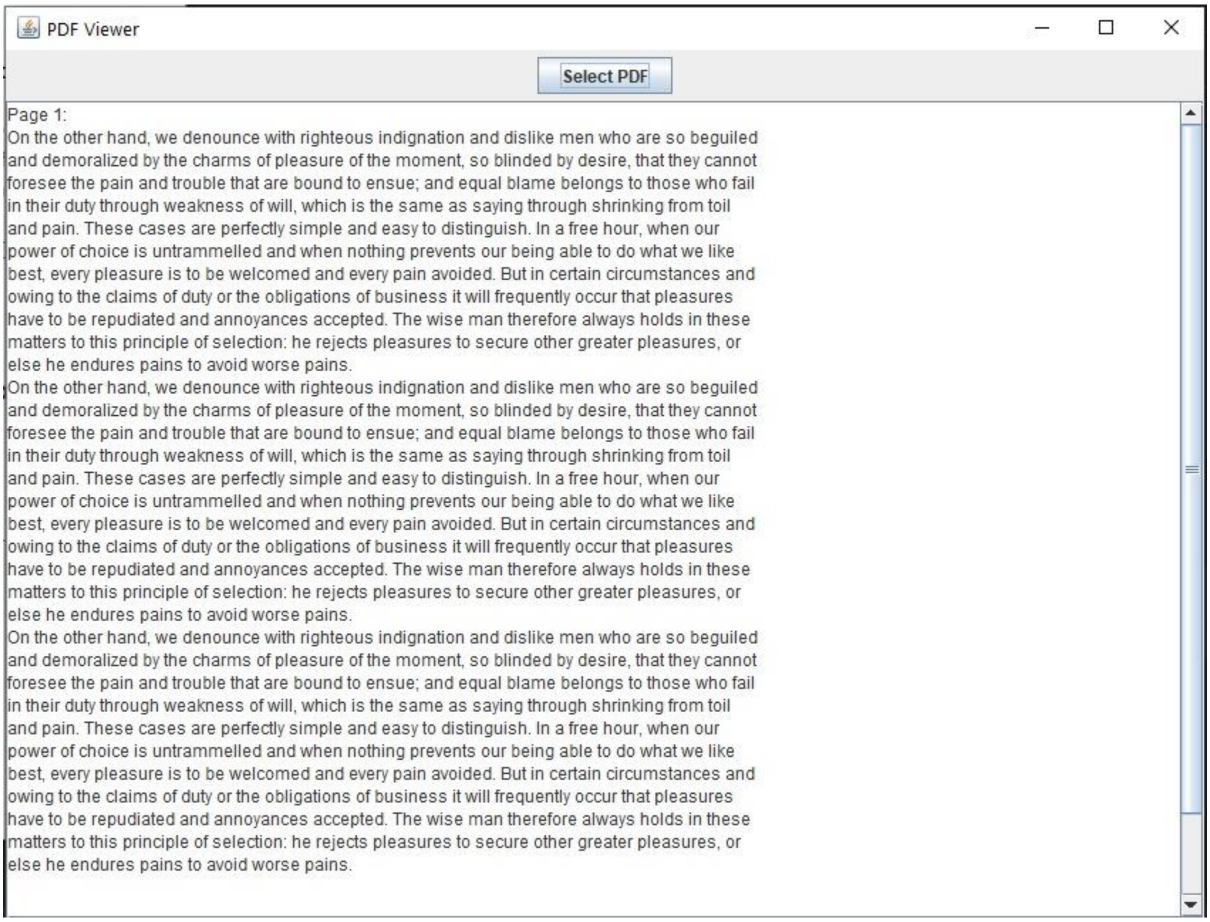




5. View PDF

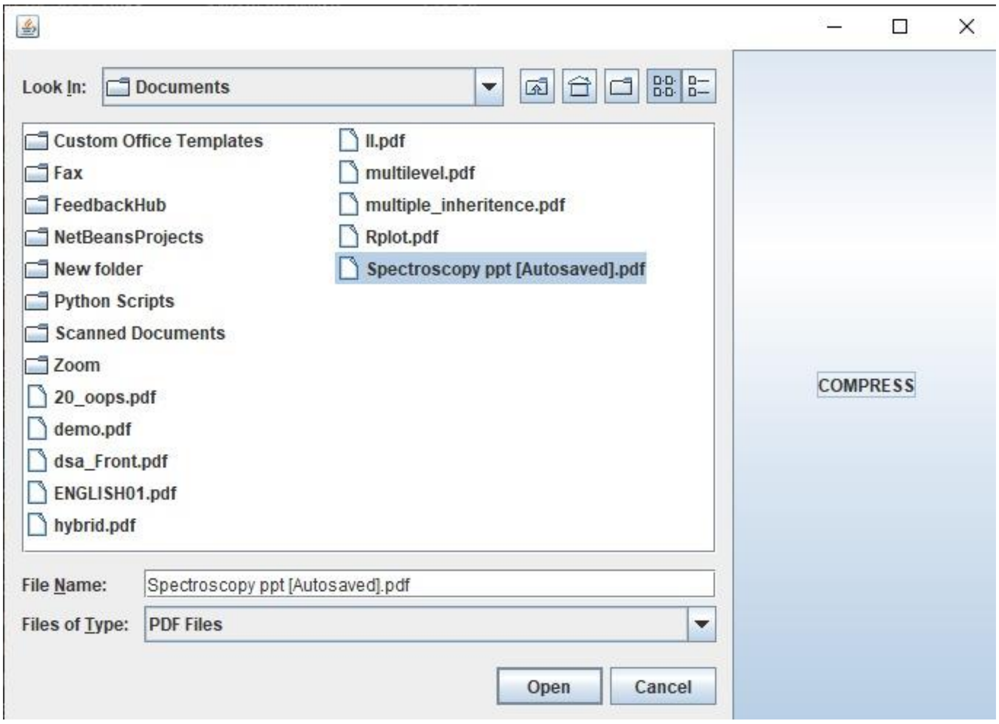
Selecting the file to open





6. **Compress PDF**

Selecting the to compress



FUTURE WORK

There are several changes that could be added in the future to improve a project. Some potential changes for a project could include:

1. **Security features**: Adding password protection can be an effective way to keep sensitive information safe from unauthorized access. It is important to implement strong password requirements to ensure that the password cannot be easily guessed or cracked.
2. **Improving GUI**: Improving the graphical user interface (GUI) can make the project more user-friendly and visually appealing. This can include changing the layout, colors, fonts, and icons to make the project more visually appealing and easier to navigate.
3. **Adding features**: Adding features such as the ability to convert PDF to Word, Word to PDF, and other related features can make the project more versatile and useful. These features can help users to easily transfer information between different file formats and make the project more valuable.

Other changes that could be considered include improving performance, optimizing code, and adding new functionality based on user feedback. It is important to keep in mind the project's goals and objectives when considering changes, and to prioritize changes based on their potential impact on the project.

CONCLUSION

In conclusion, the PDF Coalesce tool is an efficient and versatile solution for managing PDF documents. It offers several features, including merging, splitting, adding, extracting, viewing, and annotating PDF files, which makes it an essential tool for users who frequently work with PDFs. The project utilizes several key concepts, including functions and classes, exception handling, threads implementation, Swing, AWT, and the use of libraries such as pdfbox and itextpdf. Additionally, the project incorporates the Maven concept, which simplifies the process of managing dependencies and building the project. Overall, the PDF Coalesce project is an excellent example of a well-designed and implemented tool that can help users effectively manage PDF documents.