TEXT EDITOR

SCRAP BOOK

REPORT BY

ABHAY RAJ AP21110010623

DEDIPYA GOSWAMI AP21110010650

ADNAN KHAN AP21110011217

D M AKSHAY AP21110011219

Submitted for the course JAVA PROGRAMMING LAB CSE 207L

Mentored by

Prof. L Srinivasa Rao

Acknowledgement

To Professor L Srinivasa Rao Sir, we would like to offer our sincere gratitude and appreciation for their excellent advice, assistance, and knowledge while we were developing our Java project.

Their in-depth expertise, astute criticism, and unrelenting dedication to quality have all been crucial in moulding the project into what it is today. The support we received from Professor L Srinivasa Rao Sir were crucial in improving our technical abilities and widen the grasp of Java programming ideas. Furthermore, we would like to express our gratitude for their constant availability and willingness to address queries and concerns. Their guidance has not only enhanced my technical abilities but also fostered personal and professional growth.

Abstract

The documentation provides a comprehensive overview of the system architecture, design considerations, and implementation details. It includes a detailed description of the functionalities, algorithms, and data structures utilized in the text editor. Additionally, user instructions and guidelines are provided to ensure a seamless user experience. Furthermore, the text editor incorporates a customizable user interface, allowing users to personalize their editing environment by selecting themes, adjusting font preferences. This flexibility enables users to tailor the editor to their specific preferences and workflow.

Description

The goal of this project is to provide a feature-rich Java text editor that will make manipulating text-based documents easier. It includes fundamental features including text formatting, document creation, opening, saving, and search activities. The text editor has a customisable user interface and allows undo/redo operations. The project guarantees modularity and maintainability of the software through the use of Java's libraries and adherence to object-oriented principles.

The Text Editor project in Java aims to develop a versatile and user-friendly application that allows users to create, edit, and manage text-based documents efficiently. The project focuses on providing essential functionalities and features, coupled with a seamless user experience. The text editor also incorporates essential editing capabilities such as cut, copy, and paste, enabling users to manipulate text content effortlessly. It provides a customizable user interface, allowing users to personalize their editing environment by selecting different themes, adjusting font preferences, and configuring shortcut keys according to their preferences. Overall, the Text Editor project in Java offers a robust and feature-rich solution for efficient text editing. Its user-friendly interface, extensive functionalities, and customization options make it suitable for a wide range of users, including writers, programmers, and students, who require a reliable and versatile text editing tool.

Applications

1

Documentation

The text editor project can be used by writers, authors, and content creators for drafting and editing various types of documents such as articles, essays, reports, and manuscripts.

2

Note-taking and Documentation:

The text editor project can be used as a digital notebook or documentation tool. Users can quickly jot down their ideas, take notes during meetings or lectures, and organize information in a structured manner.

3

Personal Journaling:

The text editor can be employed as a personal journaling tool, allowing individuals to record their thoughts, experiences, and reflections. Bloggers can use the editor to write and format blog posts before publishing them online.

4

Programming and Coding:

The text editor can serve as a lightweight integrated development environment (IDE) for programmers and software developers. It allows them to write, edit, and manage source code files in different programming

System requirements

Operating System:

Windows 7 Version and higher.

Processor:

Intel i3 or above.

Ram:

System Ram of 4 GB or higher for ease of experience.

Compiler and Code Editor:

VS code is preferable.

These are minimum requirements and may vary depending on the specific version of Windows being used.

Implementation and Explanation

There are many ways through which we can access our application. The preferable method before deploying the application is to open our code editor in which all of our modules are stored. Locate the main program i.e., TextEditor.java and run it. A dialog box will pop up and we can access our text editor application.

We are provided with a white screen in which we can perform the write operation i.e., type all the content. We are also given a top bar in which 5 options provided.

The first option is “File”. In the submenu of file, we have 5 options again i.e., “New”, “Open”, “Save”, “Save as” and “exit”. These are general options that we can find in every text editor. The names imply the general functions that usual options in other text editors provide.

The next option is “Edit”. This option has 5 other options in its submenu. Those are “Undo”, “Redo”, “Cut”, “Copy”, “Paste” and “Select all”. These options help us to perform any form of manipulations we want on the text we have typed. We also have other options to reuse the text i.e., cut, copy and paste options. If we want to select all the text, we have typed we are provided with select all option.

The third option is “Font”. This option helps us to manipulate the styling and dimensions of our text. Refer output to get a better idea about this option. We also have “bold”, “Italic” and “Underline” options to add extra dynamic aspects to required texts.

The fourth option is “Color”. This option has a submenu with three options i.e., “Font color”, “Background color” and “Change theme”. These options change the interface characteristics and the way the font is presented and helps us to set our desired environment.

The last option is “Help”. This option provides all the information that a user needs to access and use Scrap Book.

Modules

We have used java programming language to develop our text editor application. In java we follow modular programming approach and use different modules to build a complicated application. We also followed the same modular programming approach to develop our text editor application.

Certainly! Here's the revised project description with subheadings for the first two points and the last three points:

- Text Editor is a Java application designed to provide a feature-rich environment for text editing and manipulation.

- The application offers a range of functionality to enhance the editing experience, making it suitable for various text-related tasks.

- Text Editing:

  - Users can create, open, edit, and save text files.

  - Support for basic text editing operations such as typing, deleting, copying, and pasting.

  - Multiple file format support and options to save files with different extensions.

- Font Styling:

  - Ability to apply font styles such as bold, italic, and underline.

  - Customization options for font size and color.

  - Enhances the visual appearance of the text.

- Undo and Redo:

  - Functionality to revert changes or redo previously undone actions.

  - Allows users to correct mistakes or experiment with different editing approaches.

- Themes:

- Ability to change the visual theme or look and feel of the user interface.

- Offers a variety of available themes to customize the appearance of the text editor.

- Cut, Copy, and Paste:

- Operations for manipulating and transferring text content.

- Provides easy content rearrangement and transfer between different sections or applications.

- Keyboard Shortcuts:

- Support for keyboard shortcuts for common operations.

- Allows users to perform actions quickly and conveniently without relying solely on mouse interactions.

- User-Friendly Interface:

- Clean and intuitive user interface design.

- Clear menus and icons for quick access to different functionalities.

- The Text Editor caters to different users, including students, writers, programmers, and individuals in need of a reliable text editing tool.

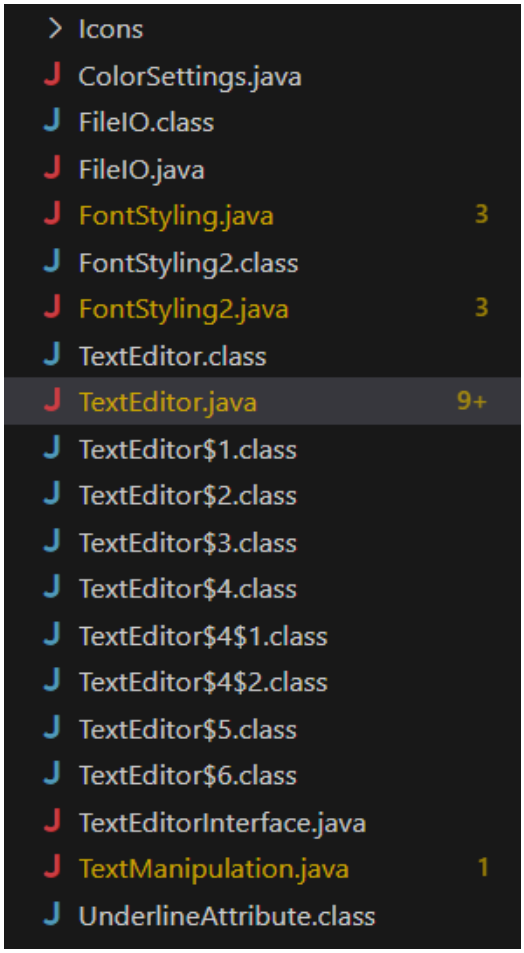
- With its versatility and user-friendly design, the Text Editor is suitable for both basic editing needs and more advanced requirements.

- Whether creating, modifying, or formatting text, the Text Editor provides a convenient and efficient environment for text manipulation.

Note: The subheadings help to categorize the features of the Text Editor, making it easier to understand and navigate through the

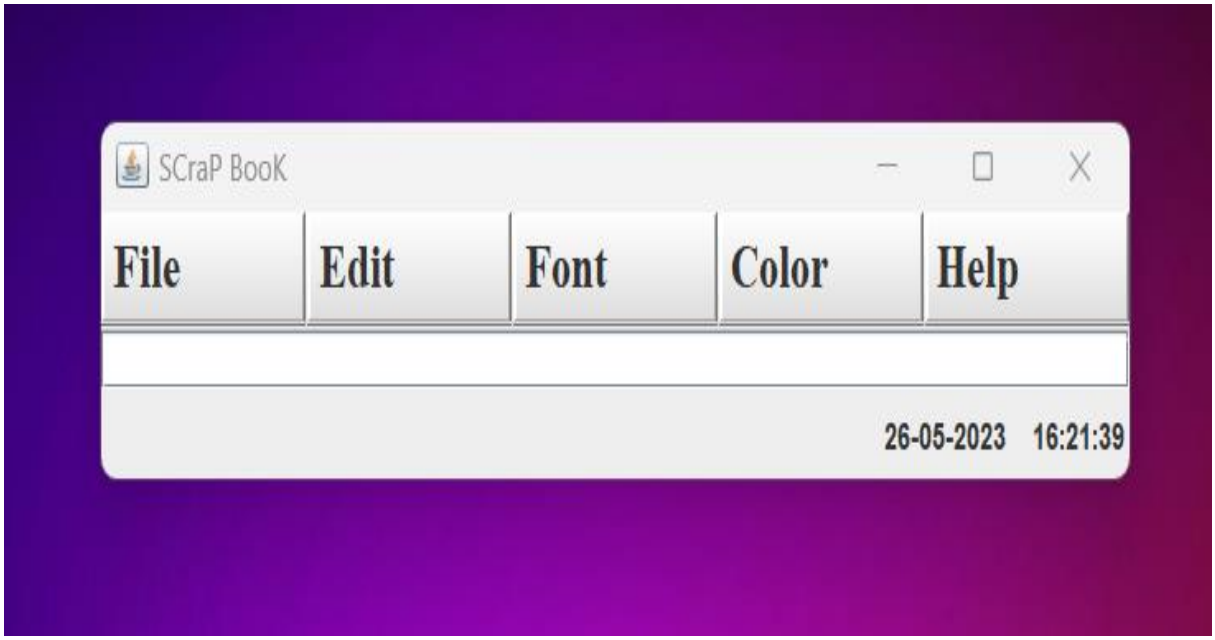
different capabilities of the application. Feel free to customize the subheadings based on the specific features and functionalities of your text editor project.

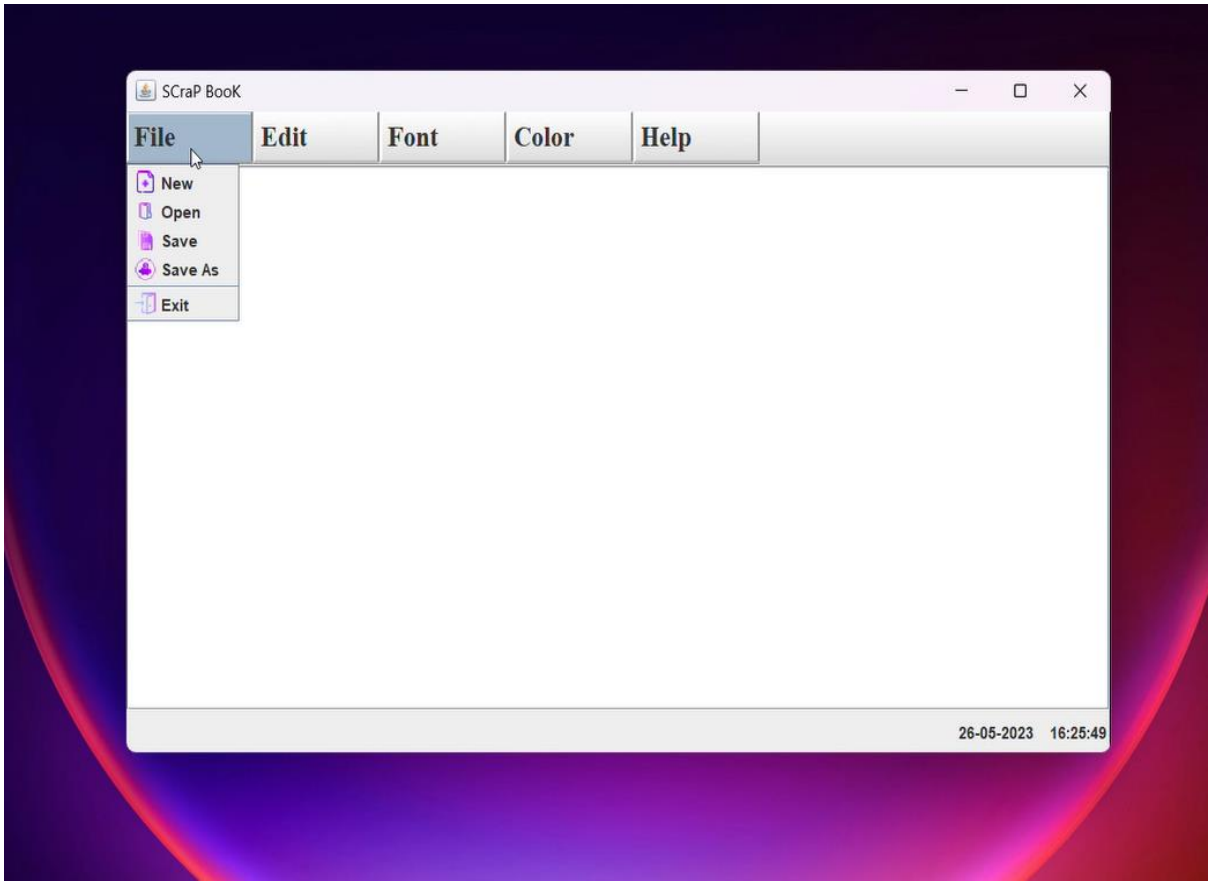
Modules we have used to implement these functions:

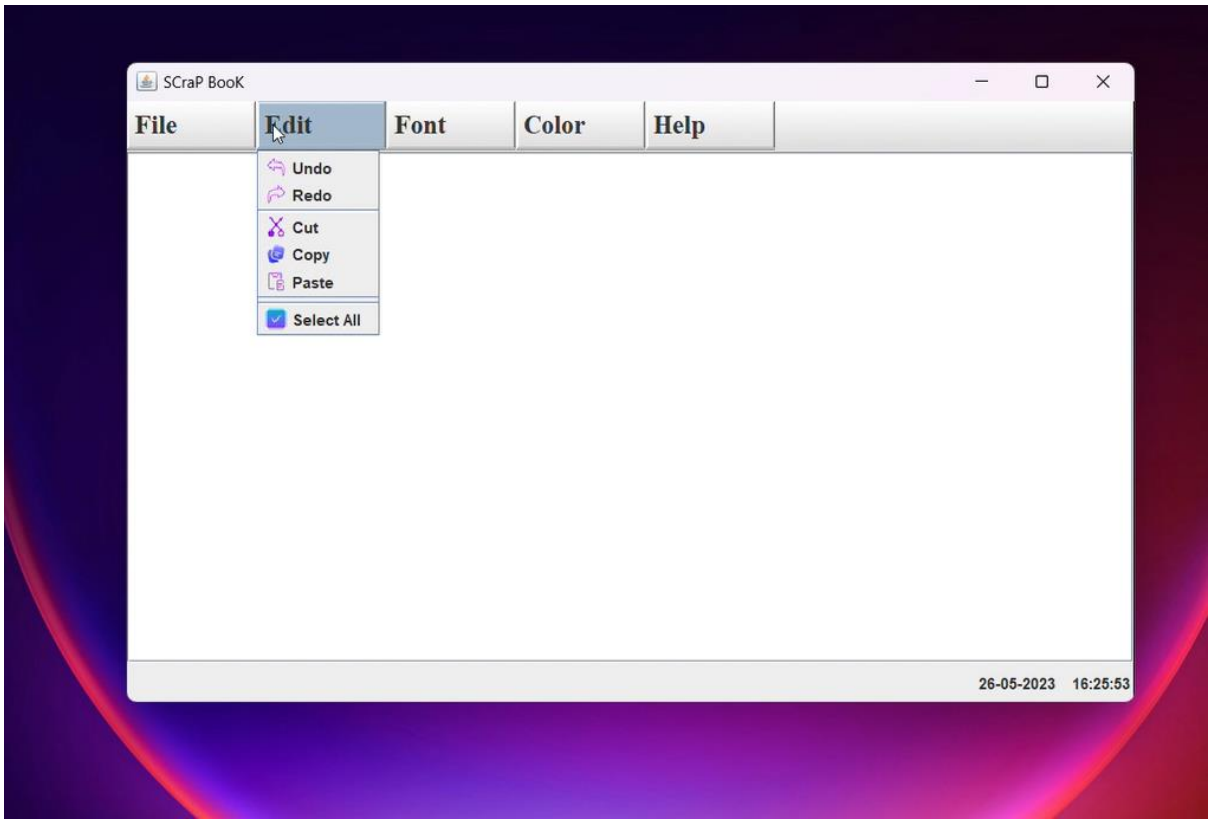


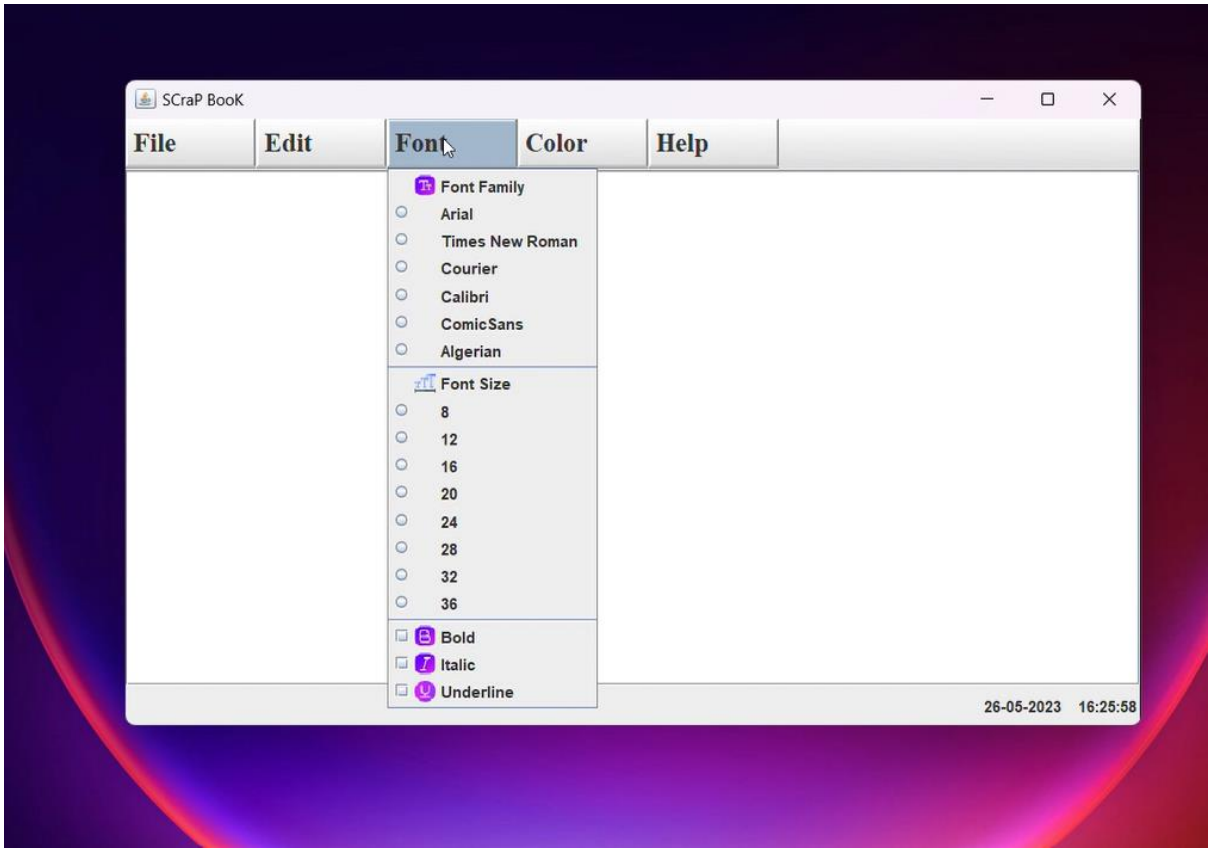
We also have a folder named “Icons” in which we have all the .png files required to display icons for each option.

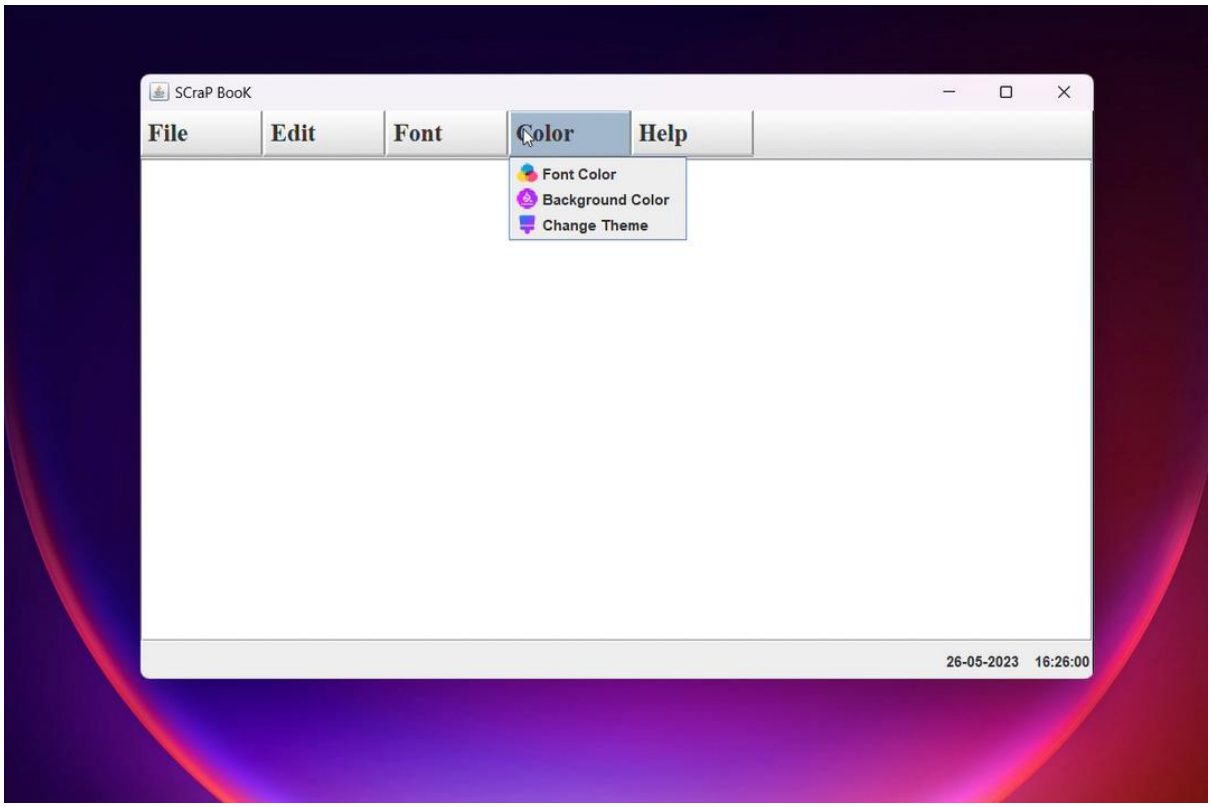
Output

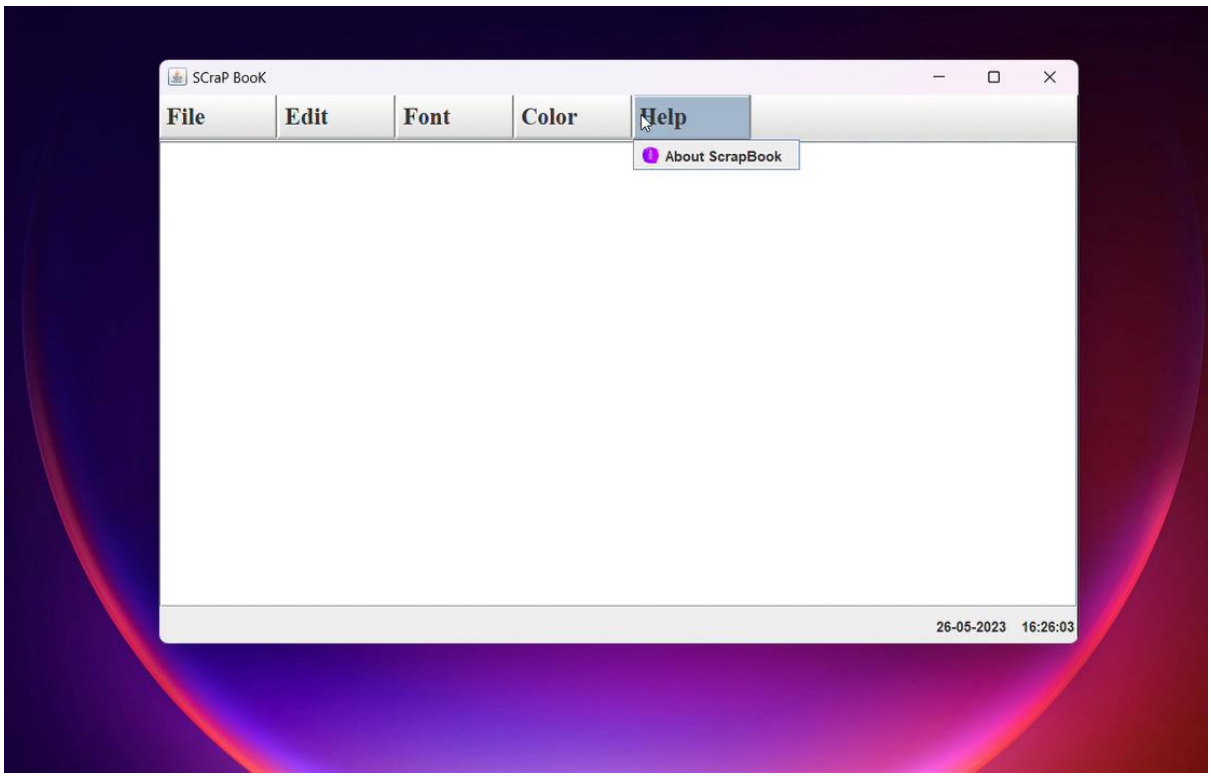

SCraP BooK
File
Edit
Font
Color
Help
26-05-2023 16:21:39


SCraP BooK
File
Edit
Font
Color
Help
New
Open
Save
Save As
Exit
26-05-2023 16:25:49


SCraP BooK
File
Edit
Font
Color
Help
Undo
Redo
Cut
Copy
Paste
Select All
26-05-2023 16:25:53


SCraP BooK
File
Edit
Font
Color
Help
Font Family
Arial
Times New Roman
Courier
Calibri
ComicSans
Algerian
Font Size
8
12
16
20
24
28
32
36
Bold
Italic
Underline
26-05-2023 16:25:58


SCraP BooK
File
Edit
Font
Color
Help
Font Color
Background Color
Change Theme
26-05-2023 16:26:00


SCraP BooK
File
Edit
Font
Color
Help
About ScrapBook
26-05-2023 16:26:03


26-05-2023 16:26:03

Future Enhancements

Future Enhancements for the Text Editor Project:

Collaboration and Version Control: Implementing real-time collaboration features would enable multiple users to work on the same document simultaneously. Additionally, integrating version control functionality would allow users to track changes, compare revisions, and revert to previous versions of the document.

Advanced Text Editing Features: Introducing advanced text editing capabilities like auto-complete, code snippets, and intelligent code suggestions can significantly enhance the text editor's functionality for programming and coding purposes. These features can improve productivity and accuracy while writing and editing code.

Integration with Cloud Services: Integrating the text editor with cloud storage services such as Google Drive or Dropbox would enable users to seamlessly save and sync their documents across multiple devices. This would enhance accessibility and data security, allowing users to work on their documents from anywhere.

Customizable Themes and UI: Enhancing the text editor's visual appeal by providing a wide range of customizable themes and user interface options would allow users to personalize their editing environment further. This could include different color schemes, font styles, and layout options to suit individual preferences.

Advanced Search and Replace Functionality: Enhancing the search and replace functionality with advanced features like regular expressions, case sensitivity options, and batch processing would provide users with more powerful and efficient search capabilities, allowing them to manipulate text in a more precise manner.

Conclusion

The Text Editor project in Java has successfully achieved its objective of developing a versatile and user-friendly text editing application. Throughout the project, we focused on providing essential functionalities and features while ensuring a seamless user experience. By incorporating features such as document creation, opening, saving, text formatting, undo/redo operations, the text editor offers a comprehensive set of tools for efficient text manipulation. The customizable user interface allows users to personalize their editing environment, enhancing their productivity and comfort.

In conclusion, the Text Editor project in Java has successfully delivered a robust and feature-rich solution for text editing. The project's accomplishments highlight the power of Java programming in developing practical applications, showcasing the benefits of a well-designed and user-centric approach.