IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, March 2026
Impact Factor: 8.2

Notepad Application Using Python

Tanishka Dhananjay Dounde[1], Tejasavi Bhaurao Kengar[2], Priya Mukesh Thaware[3], Sujata Mahadev Sanap[4]
Students, Department of Computer Engineering[1-4]
Guide, Department of Computer Engineering[5]
Rasiklal M. Dhariwal Institute of Technology, Pune, India

Abstract: *They are one of the fundamental software programs used to create, edit, and manage text-based files in a computer. These applications are commonly used by students, programmers, writers and professionals to write notes, edit source codes and write simple documentation. In this research paper, a Notepad application in Python using Tkinter library has been designed and developed. In this project, our goal is to develop a lightweight and simple Enhanced Text Editor with only the most popular text editing features which are present in basic Notepad applications. It enables users to make new text documents, access existing ones, edit and save the changes. The graph is created using the Tkinter framework that allows building interactive windows, different menus and text areas. Basic editing features like cut, copy paste and a few text manipulation options are also part of the system so a user can efficiently use the tool for their desired purpose. Python was chosen due to its high readability, simplicity and vast number of libraries that help in developing applications quickly. This implementation is a boilerplate explanation on how to build simple desktop apps with graphical interface using python. The Notepad application we developed is user-friendly, lightweight and capable of doing simple text editing operations. The project can also assist novices in learning about basic programming principles such as GUI designing, event handling, file management when coding on python.* .

Keywords: Python, Tkinter, Text Editor, Notepad Application, GUI Development, File Handling, Desktop Application

I. INTRODUCTION

Text editing applications are important for enabling the creation and management of textural information within today's digital landscape. A text editor is a basic software that enables user to create, edit and save plain text documents. These applications are typically used for writing notes, drafting documents, editing configuration files and source code. They are used everywhere such as Notepad since they provide an easy and lightweight environment to write text. They are simple but powerful utilities for daily computing tasks. Programming technologies have greatly advanced over the years, and developers are now able to develop custom text editor applications with improved and augmented functions.

Because of its easy-to-learn syntax and huge library ecosystem, Python has become one of the most popular programming languages for software development. There are many libraries in Python which allow the developers to develop graphical user interfaces (GUI) directly for their desktop applications. One common library to implement such behavior is Tkinter; it is included with the Python standard library and enables us to create windows, buttons, menus, text fields etc. Tkinter assists developers in creating interactive applications where a user does some task using graphical controls instead command line commands.

The project is to make Notepadthru Python and Tkinter library. This application allows you to implement basic text editing functionalities (like creating files, opening them, making changes in content etc) and save those contents into the files. Driven user experience and fast a simple start to build GUI apps with python. Write a desktop application with python that talks to everyday computing tasks. No other language on the.NET runtime offers as much support for functional programming.

DOI: 10.48175/568

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, March 2026
Impact Factor: 8.2

II. LITERATURE REVIEW

Many software tools and applications have come up in the years to help users manage their text files. Notepad, WordPad, and other lightweight text editors use simple features primarily as writing or text editors. These applications are part of and often included with operating systems, and are used extensively for basic documentation purposes. These editors are pretty useful and informative, but sometimes you might need more tools depending on what you are doing. Well, text editors will still be there because they are light and easy to use.

Python has emerged as one of the popular programming languages in software development for developing applications owing to simplicity and high flexibility. Also, It supports multiple programming paradigms along with a huge array of libraries helpful in development. Python works great for developing GUI Applications as well, with libraries Tkinter, PyQt and Kivy supporting it. This is generally regarded as one of the easiest options for building a tool if you have never had experience creating a GUI.

Tkinter has been used in building small desktop applications like calculators, note taking tools or simple text editors as it was shown in some previous projects and studies. Note that none of these use very fancy programming techniques and Python is actually well-suited to creating interactive bits of software. This project extends these ideas by implementing a Notepad application in Python that offers basic text editing capabilities. This system reflects a practical example of Python programming and graphical user interface development by its focus on simplicity, easiness to use, and optimization.

III. RESEARCH GAP

Although we already have many text editors like notepad, wordpad, advanced code editors etc., mainly they provide basic functionalities for text editing or programming functionality. Normal text editors are minimalistic and lightweight but not very advanced with features like translation, encryption of texts, keyword error-checking for syntax errors in code snippets and tools to provide analytical information about the code. While modern code editors offer tons of features, they are also complex and require more system specs; hence, they might not be the best option for beginners or users who need a basic text editor environment.

This study fills these gaps by Introduces a proactive AI based scam engagement system that avails actionable intelligence. Other limitations of existing lightweight text editors are they don't offer intelligent features integrated into the editor themself. Basic editors often lack capabilities like text to speech integration, word frequency highlighting, and built-in code formatting and syntax checking. Currently, users must use many external tools to function in these areas which is inefficient and less convenient.

So a simple yet powerful text editor is necessary which will cover more features than a regular editor. On the other hand, though there are many subsequent websites-based notepad applications but they do not take into account encryption, translation features and highlighting errors or to some extent updating text analysis.Therefore, this is the progressive version of a simple python-based Notepad application using Tkinter library. It allows a more efficient and accessible option for users who need simple yet powerful desktop text editing applications.

IV. METHODOLOGY

This section describes how the Notepad application built on Python was implemented. This app is developed using the programming language Python and the Tkinter Graphical user interface library. This is a hands-on guide on how to use the API to add basic functionalities of a text editor application: create new files, open existing files, edit the texts in those new or opened files and save documents. The process of development consists of requirement analysis, system design, implementation and testing.

Requirement-Analysis

During this stage, the application is recognized with its fundamental requirement. The System Should Enable File Operations Basic functionality of the system offered should allow creating, editing, opening and saving text files. These are in addition to often used editing functions like cut, copy and paste that can contribute to better usability.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, March 2026
Impact Factor: 8.2

SystemDesign

The system design dictates the application structure. The system consists of three components-graphical user interface, application logic and file handling module. These elements work together to perform text editing tasks.

Implementation

This application employed Python and the Tkinter library. It has a menu bar and a text area where users may enter and modify text. Types of functions: File handling functions helps you to open and save Text documents.

Testing

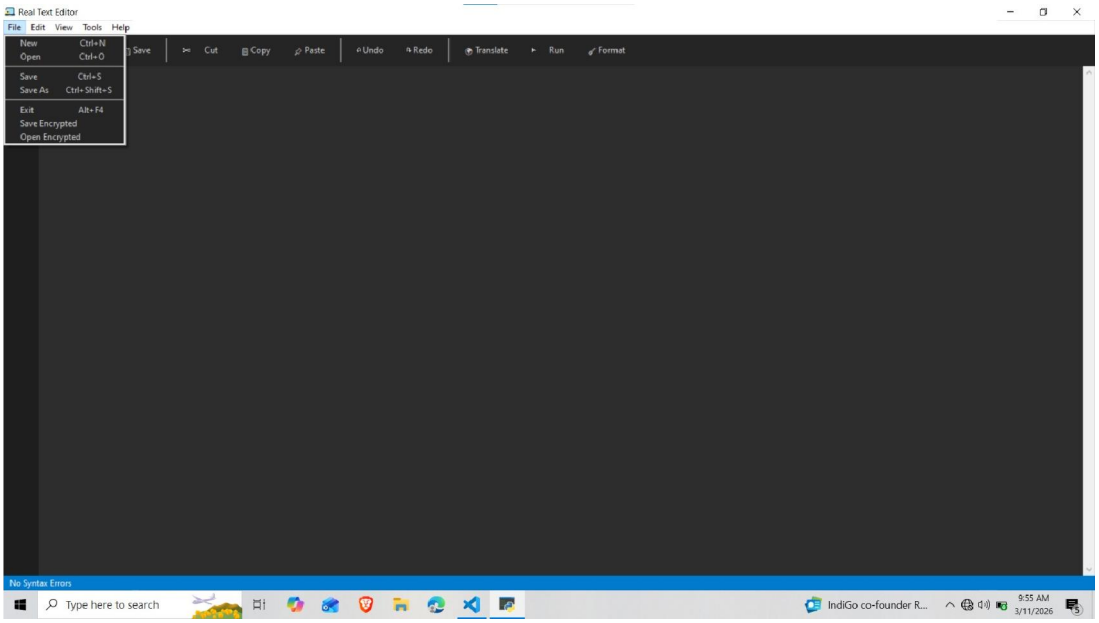
The system is tested to make sure all features work as intended. During the testing process, try to create new files, Edit text, open existing files and save documents without an error.

V. RESULTS AND DISCUSSION

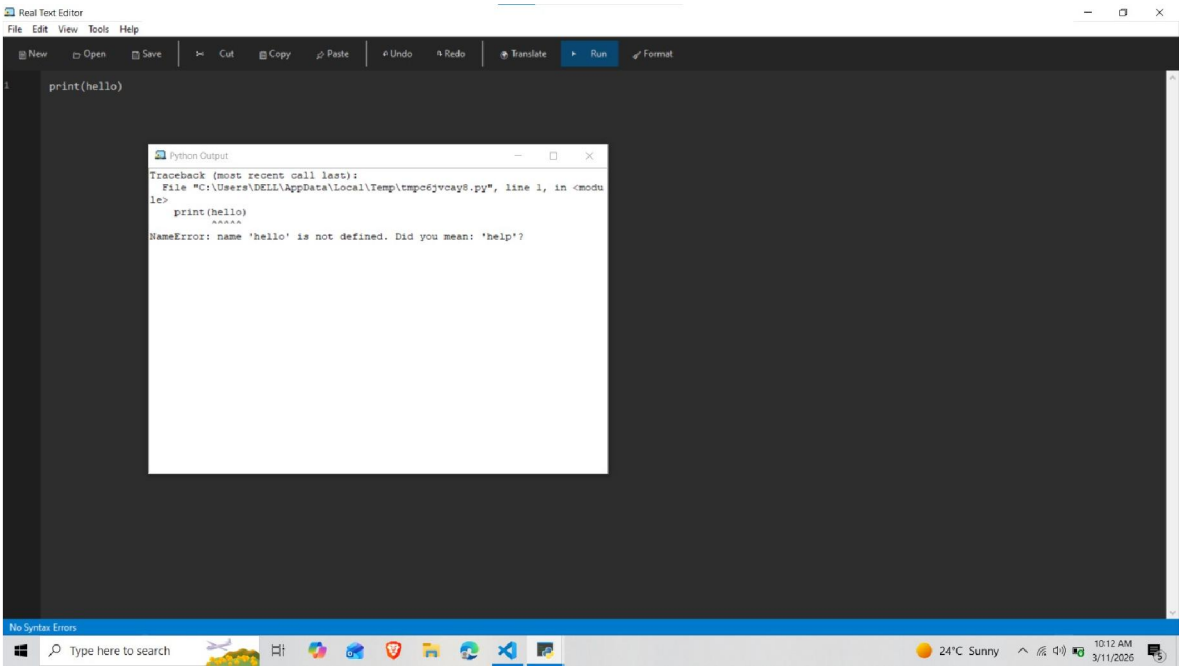
The resulting Notepad application successfully provides the necessary functionality for text editing. They offer a graphical user interface that is simple and interactive so the user can create text documents easily with graphic display notes. The application enables users to open existing files, edit the content and saves the changes without any hassle. Using Tkinter library allows for interactive UI design and seamless ongoing interaction with the program while using it.

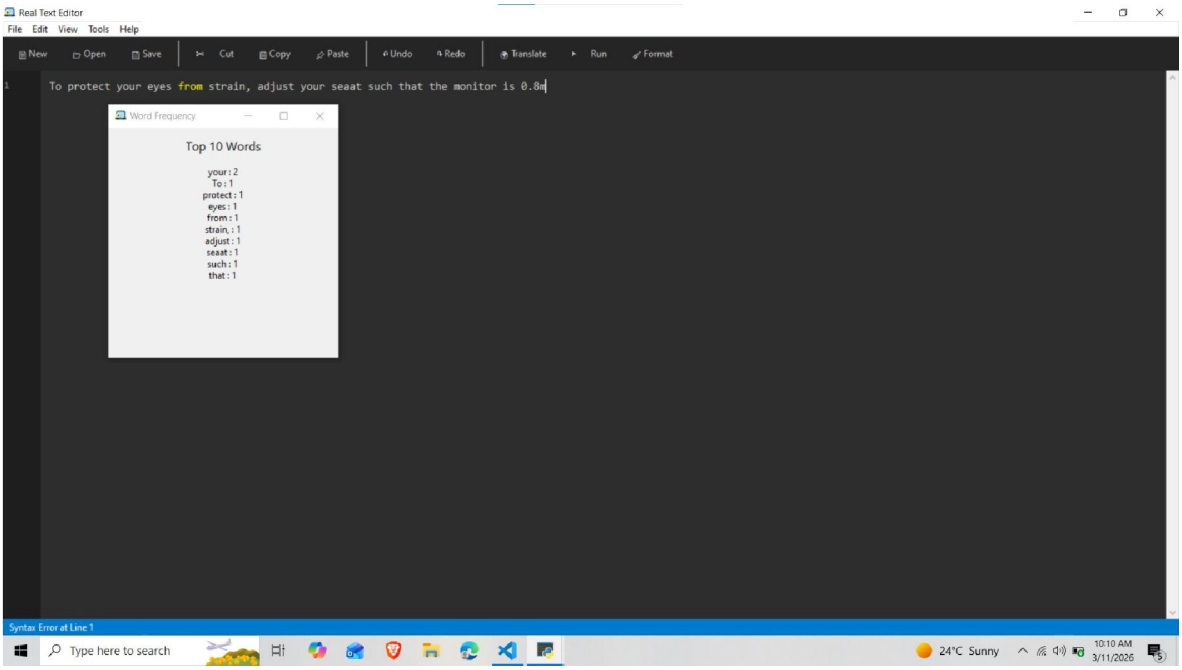
It is probable that the app works fine, the system works great for simple text editing applications. The regular command text menu options allow for speedy access to frequently used commands, and typing and editing text is seamless in the text area. Input: encapsulate the ideas of scientific advancement, technology development, and technological transfer through industrial experience

This project work shows that graphical user interface desktop applications can also be developed using Python. Python make it easier to code and Tkinter give us flexibility to write relevant software with less complexity. Then this project also focuses on the importance of user-friendly interface design in designing applications that are accessible and easy to use for a beginner too.



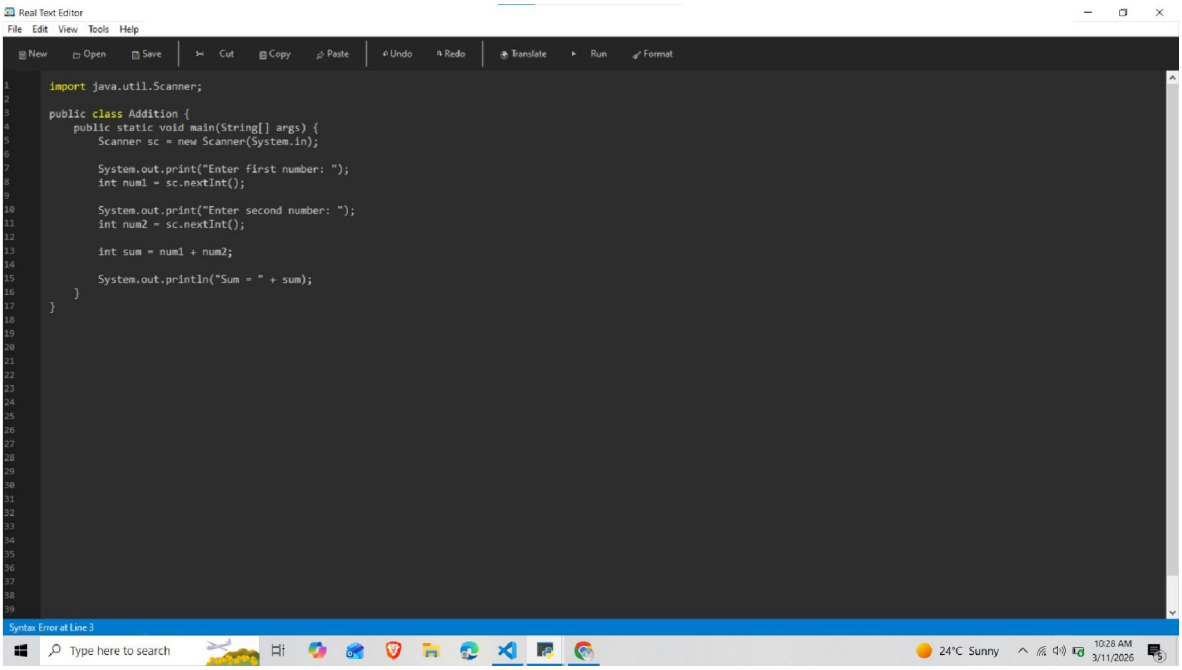
IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, March 2026
Impact Factor: 8.2





DOI: 10.48175/568

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, March 2026
Impact Factor: 8.2



VI. CONCLUSION

Creating a simple Notepad application in PythonLanguage and the Tkinter GUI libraryApplication Writing desktop applications with python/table of contentsPrevention*Table of Contents That Will Automatically Open: SQLite / Hello World! So basically this is a text editor app in which you can create new files, edit them and save documents. Such functionalities allow users to carry out basic text-related operations much faster.

One of the notable benefits you gain with this project is that the design and implementation are pretty straightforward. The fact that it is built in Python using Tkinter means the application is fairly lightweight and readable, which helps newcomers who are developing software by example. It also serves as practice for the developers to make a graphical interface, events handling and file working.

However, in general, the key goal of developing a simple Notepad application that showcases Python's potential for desktop applications is achieved. This project serves as a basis for future works that intend to create more complex text editors with additional functionalities and improvements.

VII. ACKNOWLEDGMENT

The information presented in this paper is prepared with the support and guidance received during the development of this project. Similar to On this occasion, we wish to express our sincere thanks to our project guide and the faculty members of Department of Computer Engineering are also a part of this paper for: their valuable guidance, suggestions and expert knowledge during development period.

We also want to thank our institution for the resources, facilities and academic support needed to conduct this research work successfully.

REFERENCES

[1]. Guide to Python, the (pp. 1–9). Python 3 Reference Manual. CreateSpace Publishing, USA..

[2]. Mark Lutz. (2013). Learning Python (5th Edition). O'Reilly Media, Inc.

[3]. Grayson, J. E. (2000). Python and Tkinter Programming. Manning Publications

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, March 2026
Impact Factor: 8.2

[4]. Summerfield, M. (2010). Python 3 Programming: A Complete Introduction To The Python Language Addison-Wesley.
[5]. Lundh, F. (1999). An Introduction to Tkinter. PythonWare.
[6]. Pilgrim, M. (2009). *Dive Into Python 3*. Apress Publication
[7]. Downey, A. B. (2015). Developer Guide » Think Python: How to Think Like a Computer Scientist O'Reilly Media.
[8]. McKinney, W. (2018). Python for Data Analysis. O'Reilly Media.
[9]. Python Software Foundation. *Python Documentation*. Available: https://docs.python.org
[10]. TkDocs. Tkinter GUI Programming Documentation. Available: https://tkdocs.com