CHAPTER I
INTRODUCTION

1.1 General

A text editor is a type of program used for editing plain text files. Such programs are sometimes known as "notepad" software. Text editors are provided with operating systems and software development packages, and can be used to change configuration files, documentation files and programming language source code.

A plain text file uses a character encoding such as UTF-8 or ASCII to represent numbers, letters, and symbols. The only non-printing characters in the file that can be used to format the text are newline, tab, and form-feed. Plain text files are often displayed using a monospace font so horizontal alignment and columnar formatting is sometimes done using space characters.

The core data structure in a text editor is the one that manages the string (sequence of characters) or list of records that represents the current state of the file being edited. While the former could be stored in a single long consecutive array of characters, the desire for text editors that could more quickly insert text, delete text, and undo/redo previous edits led to the development of more complicated sequence data structures.

1.2 Problem statement

To create a Text Editor using the concepts of Python programming, making use of Tkinter, pyautogui and other packages.

CHAPTER II
DESIGN AND IMPLEMENTATION

2.1 Description

The Text Editor can be used to edit and open documents. Options to change the font and theme are also included.

2.2 Requirements

1. Python3
2. Tkinter package
3. Pyautogui package

2.3 Designing User Interface

The GUI includes following facilities:

1. Text Area
2. Open files
3. Save as option
4. Change themes
5. Text editing options

2.4 Coding

Coding is done in Python importing the facilities from required packages. Following are the packages used:

- Tkinter:
  Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI. Tkinter is included with the standard Microsoft Windows and Mac OS X install of Python.

- PyAutoGUI:
  PyAutoGUI is a Python module for programmatically controlling the mouse and keyboard. The purpose of PyAutoGUI

is to provide a cross-platform Python module for GUI automation for human beings. The API is designed to be as simple as possible with sensible defaults. It can simulate moving the mouse, clicking the mouse, dragging with the mouse, pressing keys, pressing and holding keys, and pressing keyboard hotkey combinations.

- OS module:

  The OS module in Python provides a way of using operating system dependent functionality. The functions that the OS module provides allows you to interface with the underlying operating system that Python is running on – be that Windows, Mac or Linux.

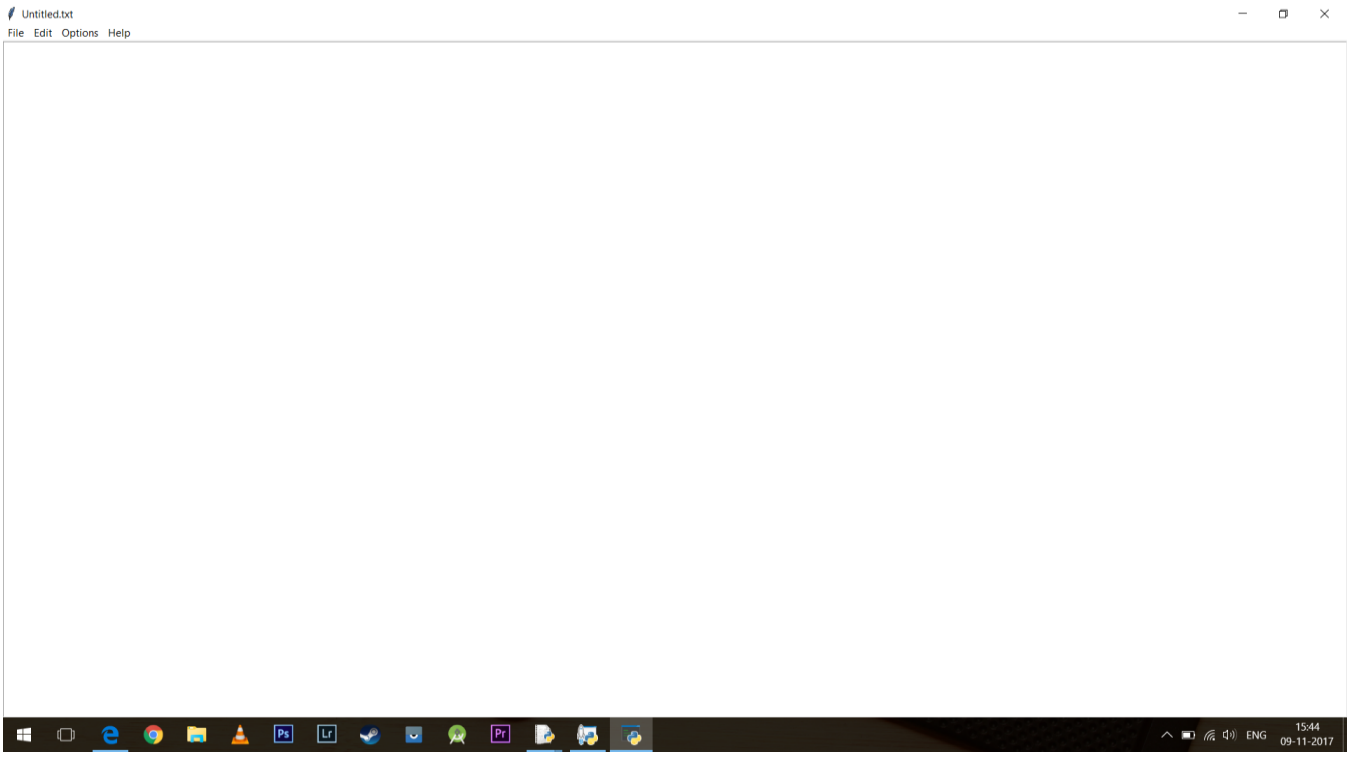
- Subprocess module:

  The subprocess module allows you to spawn new processes, connect to their input/output/error pipes, and obtain their return codes. This module intends to replace several older modules and functions.

CHAPTER III
OUTPUTS AND RESULTS

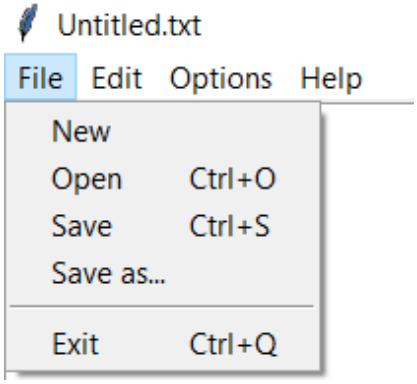
3.1 The Editor Page

This page has the facility to type text documents. It can be used to take notes and make other documents in the .txt format.



3.2 The File menu

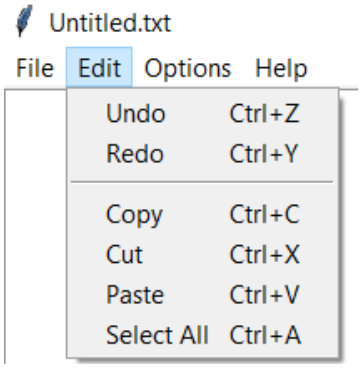
It provides options to create a new file, open an existing file, save, save as and the exit option.



New opens another tab where text can be entered.
Open will prompt the user to select a file to be opened.
Save will save the current file.
Save As will prompt the user to enter the name for the file to be saved along with its format.

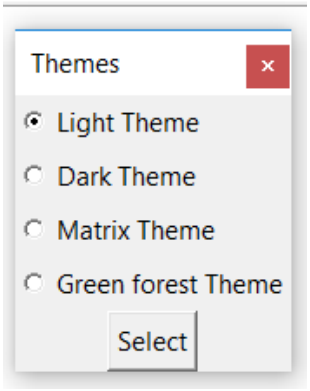
3.3 The Edit menu

It provides the editing options for a text file with functions such as undo, redo, copy, cut, paste, and select all.

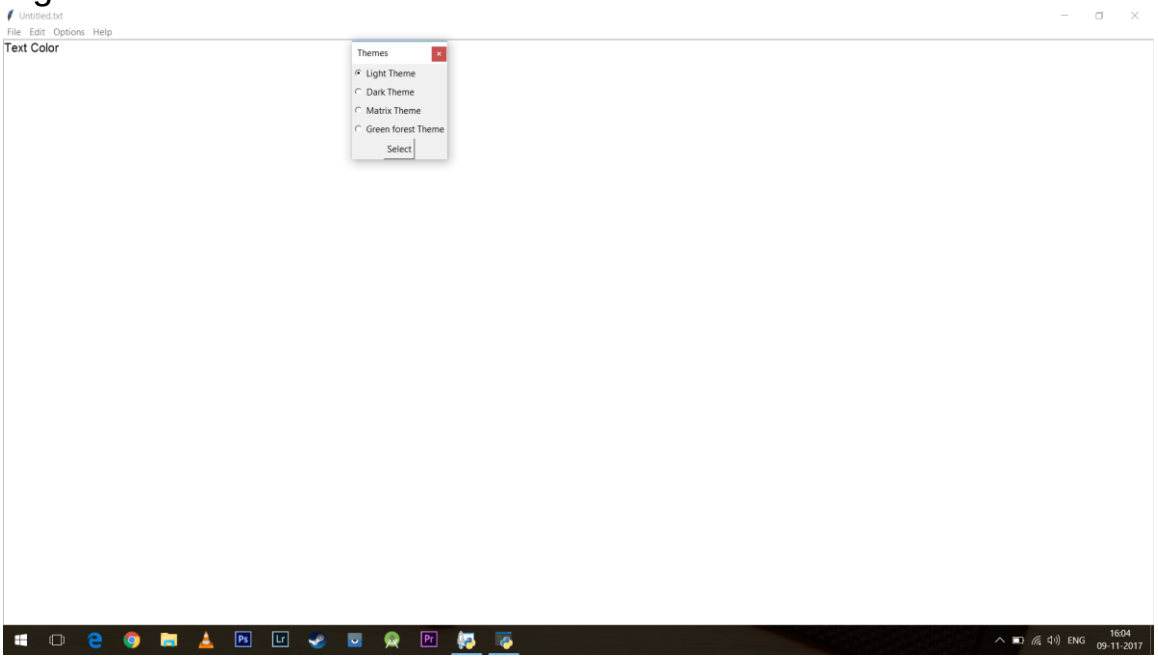


3.4 Options

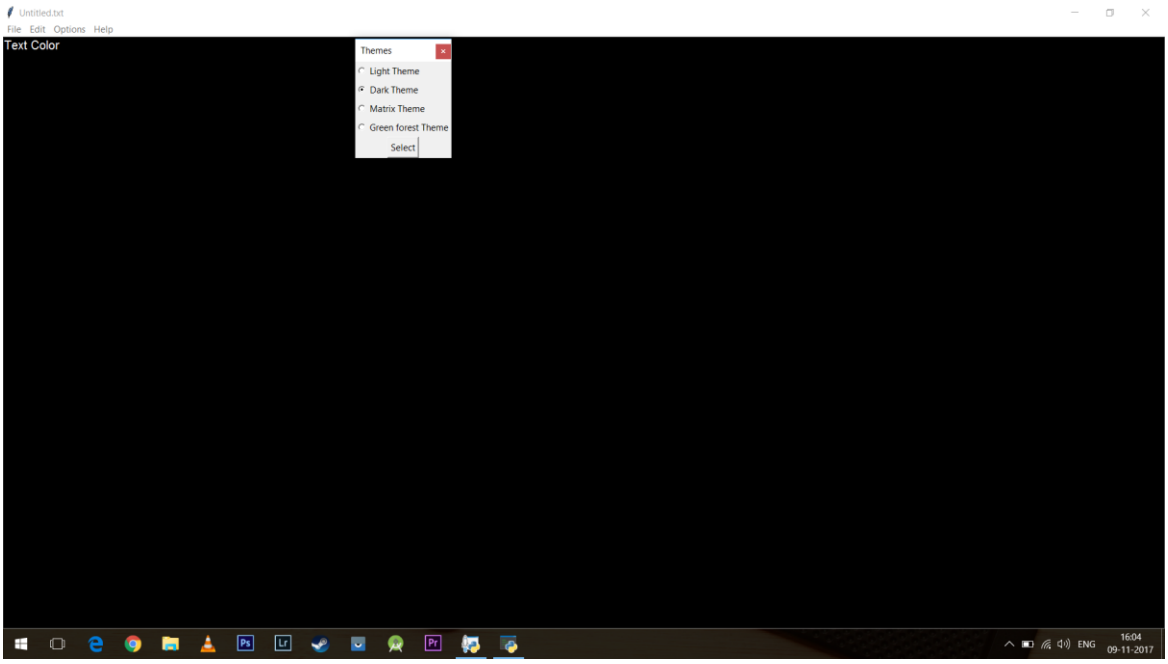
It provides the facility to manipulate the themes of the editor.



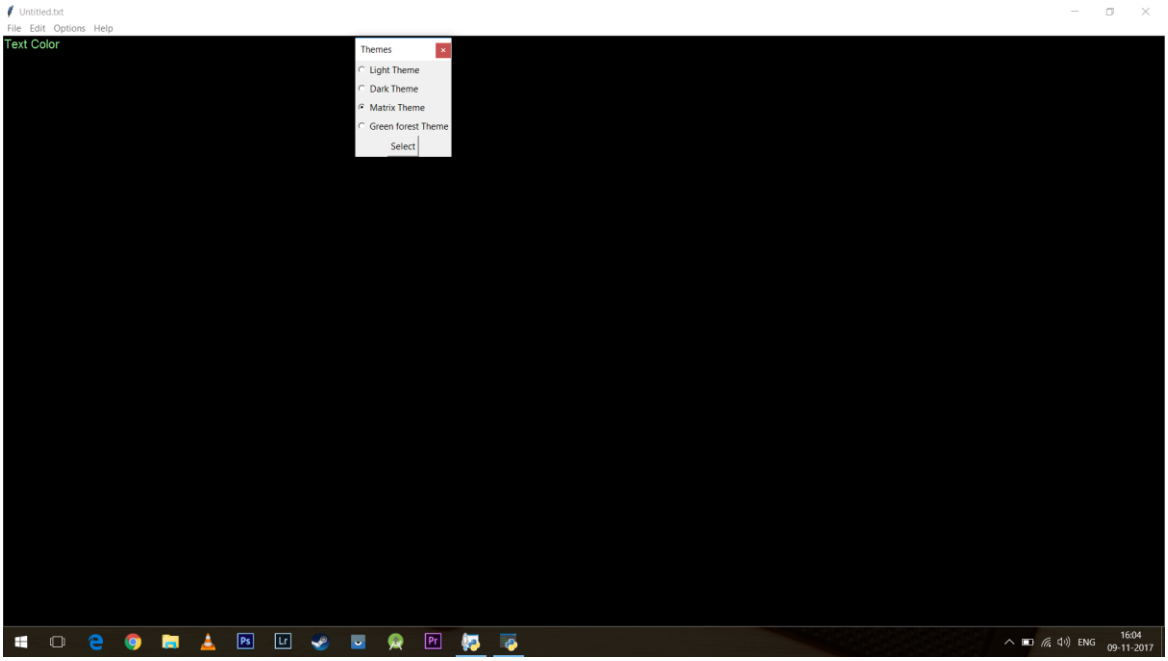
- Light Theme:



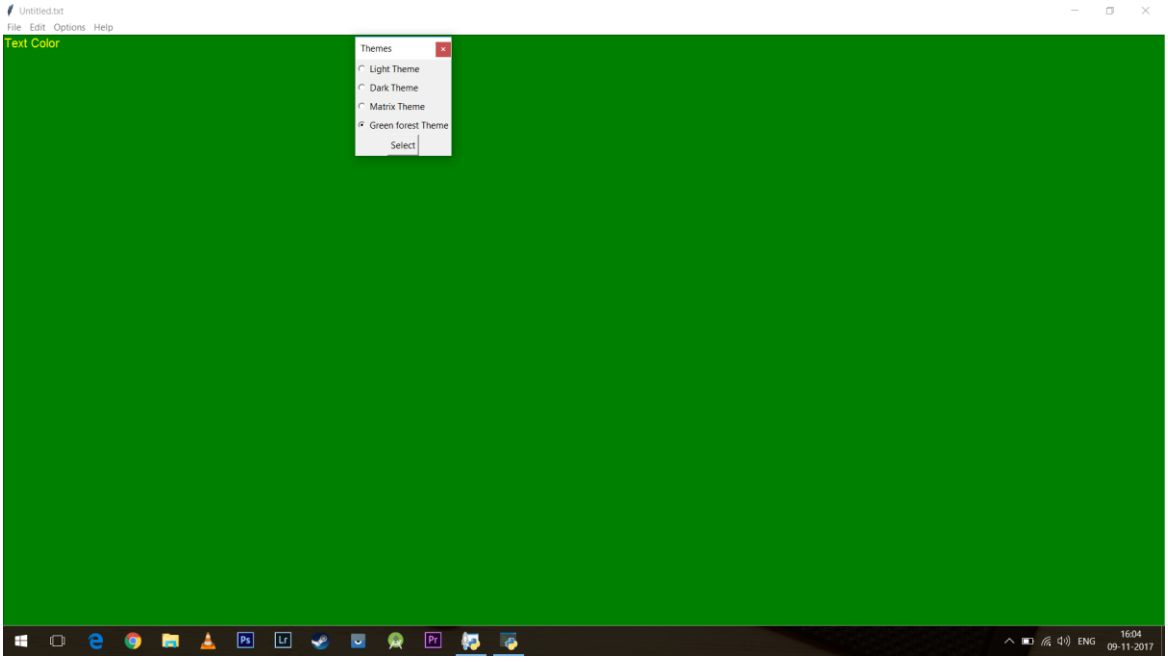
- Dark Theme:



- Matrix Theme:



- Green Forest Theme:



CHAPTER IV
CONCLUSION AND SCOPE

4.1 Special Features

Use of themes:

The text editor consists of four themes, namely, Light, Dark, Matrix, Green Forest. The theme can be changed under the options menu as per the requirements of the user. Dark theme and the Matrix theme can help save battery.

4.2 Future scope

This application can be enhanced further to add more functionalities that are not included. New fonts can be added to enhance the looks of the document that is created.